#SHAREorg

SHARE
Technology · Connections · Results

VSAM Overview

Michael E. Friske
Fidelity Investments

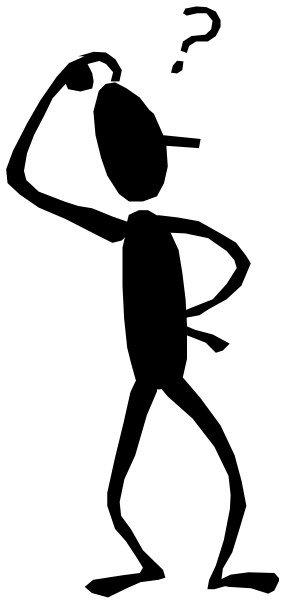
Session 11681

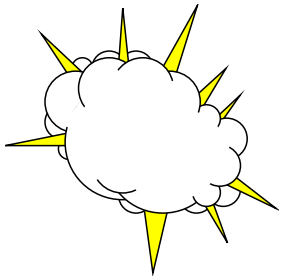
SHARE in Anaheim 2012

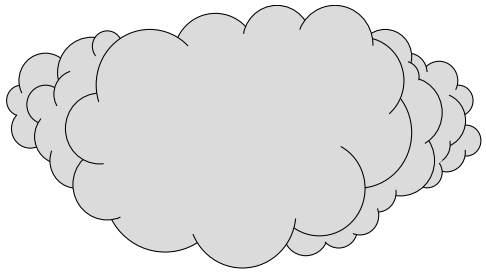
This Is a VSAM Overview Session

- This session is intended for those who know very little or nothing about VSAM.
- I will provide some basic information about defining and using VSAM data sets, but I will not go into a lot of details.
- A typical VSAM class lasts 4 – 5 days, and I only have 1 hour!

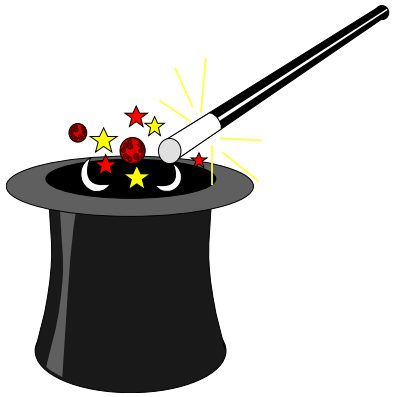
Exactly What Is VSAM?



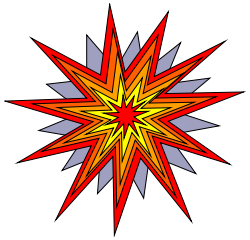




Is it a mysterious black cloud?



Is it magic?



Does VSAM Confuse, Frustrate, or Overwhelm You?

VSAM - The Acronym

V irtual

S torage

A ccess

M ethod

Types of VSAM Data Sets

- ESDS – Entry-Sequenced Data Set
- RRDS – Relative Record Data Set
- KSDS – Keyed-Sequenced Data Set
- LDS – Linear Data Set

Entry-Sequenced Data Set

- Sequences records by the order they are entered
- Supports variable length records
- Records are only added to the end of the data set
- Records can be updated, but not lengthened
- Existing records cannot be deleted, only marked as “inactive”
- Alternate index can be defined over an ESDS
- Used for applications that require only sequential access (direct access by RBA only)

Relative Record Data Sets

- Records can be either fixed-length or variable-length, but must be specified on DEFINE
- Records are stored in “slots” for fixed-length RRDSs
- Spanned records are not allowed
- Best for applications that require direct access only

Keyed-Sequenced Data Set

- Records are stored in ascending, collating sequence by key
- Records can be accessed with sequential, skip-sequential, or direct access easily with high-level languages
- Alternate indexes can be defined over a KSDS

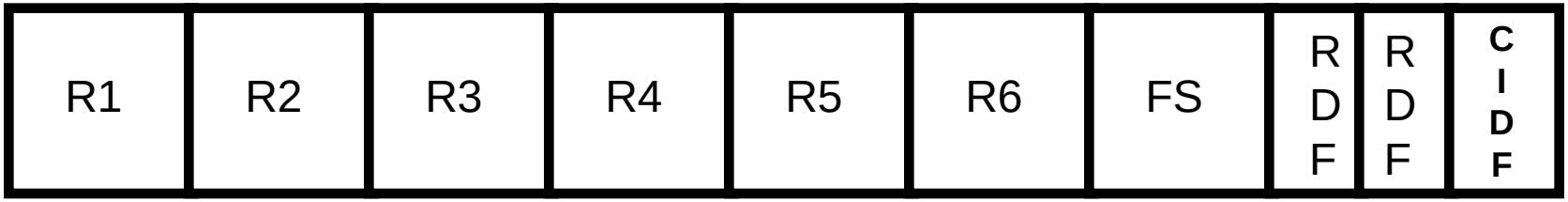
Linear Data Sets

- LDS's can be access using:
  - VSAM
  - Data-In-Virtual (DIV), if CI size is 4096
  - Window Services, if CI size is 4096
- VSAM does not have any concept of a "record" in an LDS
- DB2 is currently the biggest user of Linear Data Sets
- Many program products use Linear Data Sets

Control Interval (CI)

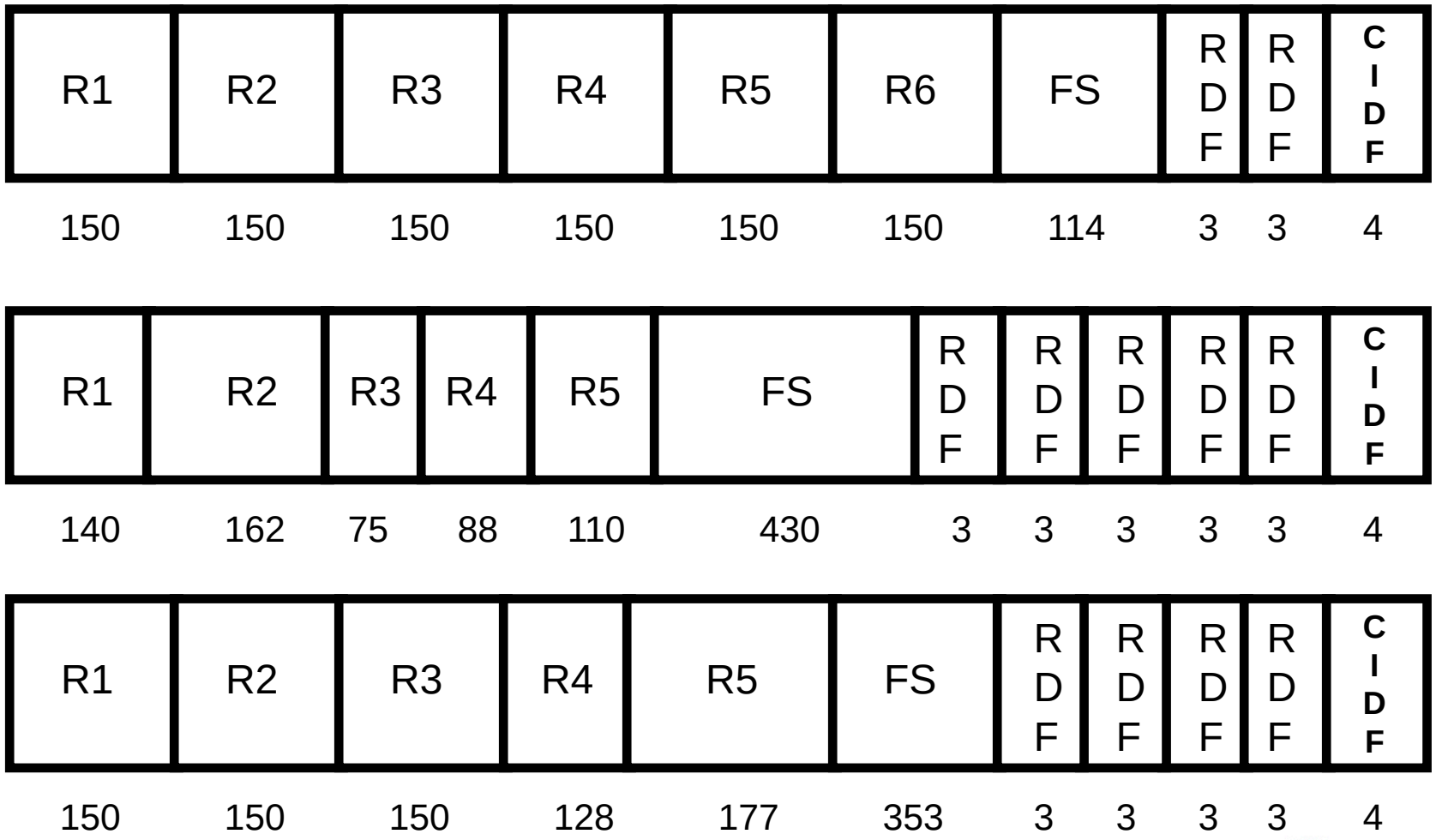
- Control Intervals (CI) in VSAM are equivalent to blocks for non-VSAM data sets
- Control Intervals contain:
  - Logical records
  - Free space
  - Control information fields
- The smallest unit of transfer between a disk and the CPU is a Control Interval

Anatomy of a Control Interval



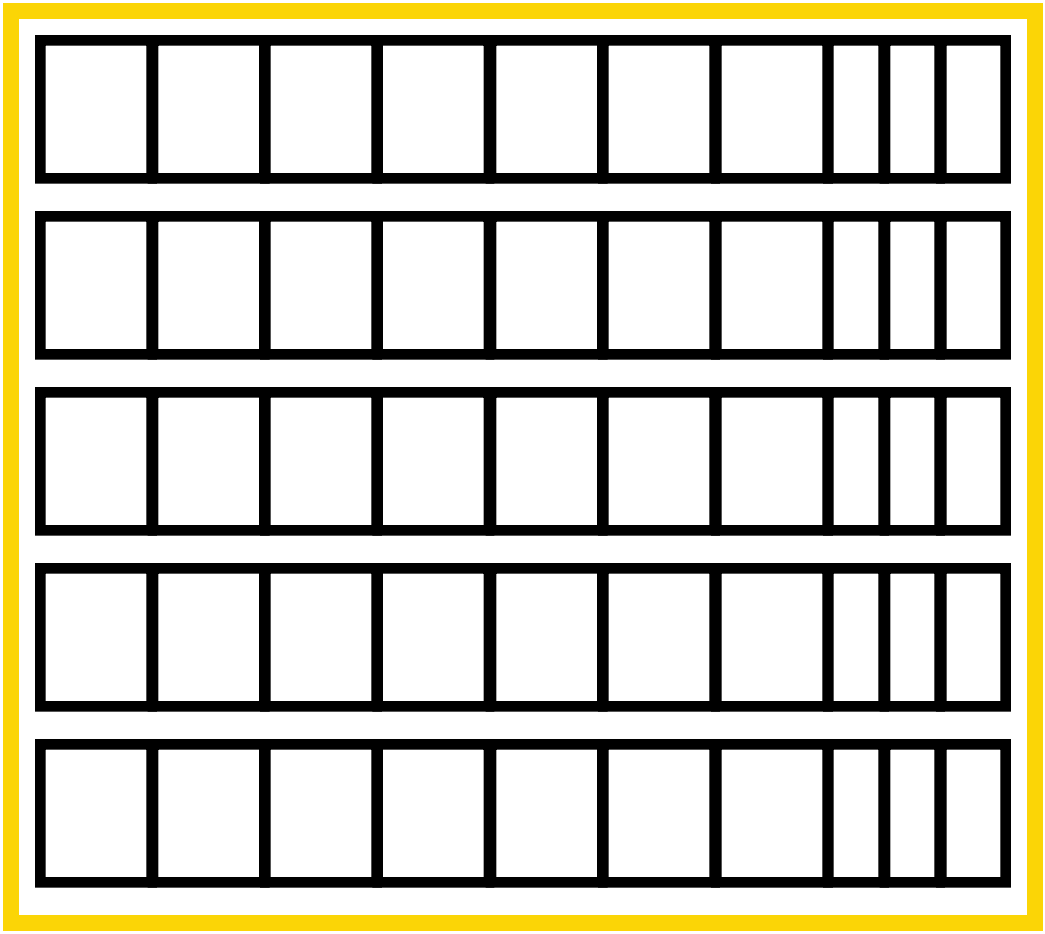
- Rn - Records
- FS - Free space
- RDF - Record Definition Fields are 3 bytes long and describe the length of records and how many adjacent records are the same length
- CIDF - Control Interval Definition Fields are 4 bytes long and contain information about the CI

How Record Lengths Affect the Number of RDFs



Control Areas (CA)

- Groups of Control Intervals (CI)
- Contiguous space on disk
- Maximum size for non-striped data sets is 1 cylinder
- CA size for striped data sets is # stripes in tracks



Size of Control Area

- The Control Area is the smaller of your primary and secondary allocation up to a maximum of 1 cylinder for non-striped data sets
- Examples:
  - TRACKS(10 2) = 2 track CA
  - CYLINDERS(10 2) = 1 cylinder CA
  - MEGABYTES(50 10) = 1 cylinder CA
  - TRACKS(100 20) = 1 cylinder CA
  - TRACKS(1 100) = 1 track CA

Advantages & Disadvantages of a Large CA

- Advantages
  - Less frequent CA splits
  - Less index records
  - More CI's can be read into storage at one time
- Disadvantages
  - More data has to be moved to do a CA split
  - More real storage and buffers are tied up during sequential operations

Defining VSAM Data Sets Using IDCAMS

```
//STEP010  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER( -
      NAME(A.B.C) CYLINDERS(50,10) SPEED -
      RECORDSIZE(150,476) INDEXED KEY(18,0)
      SHAREOPTIONS(1,3)) -
    DATA(CONTROLINTERVALSIZE(4096))
```

```
//STEP010  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER( -
      NAME(A.B.C) DATACLASS(VS150KS))
```

Defining VSAM Data Sets Using JCL

```
//NEWVS  DD DSN=A.B.C,DISP=(NEW,CATLG),
//    SPACE=(CYL,(50,10)),RECORG=KS,
//    LRECL=150,KEY=18,KEYOFF=0
```

```
//NEWVS  DD DSN=A.B.C,DISP=(NEW,CATLG),
//    DATACLAS=VS150KS
```

SMS Constructs for VSAM

- DATACLASS(dcname)
  - Can be used to specify many of the attributes needed to define a VSAM data set
- MANAGEMENTCLASS(mcname)
  - Is used to determine how to manage data sets once they become inactive
- STORAGECLASS(scname)
  - Can be used to specify the desired performance of a data set
  - Can also specify whether a data set should be defined as a striped data set

Specifying the Type of VSAM Data Set

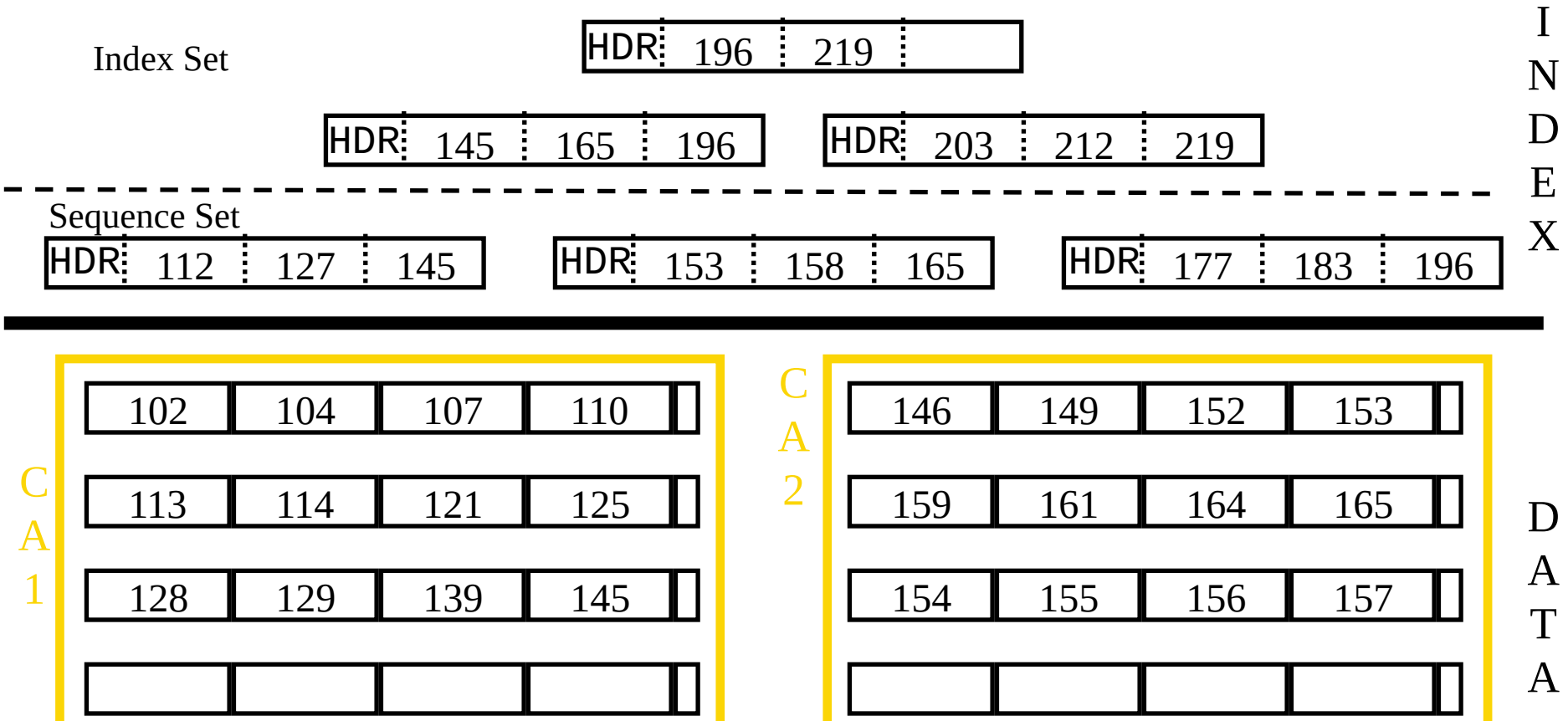
With IDCAMS

- INDEXED - KSDS
- NONINDEXED - ESDS
- NUMBERED - RRDS
- LINEAR - LDS

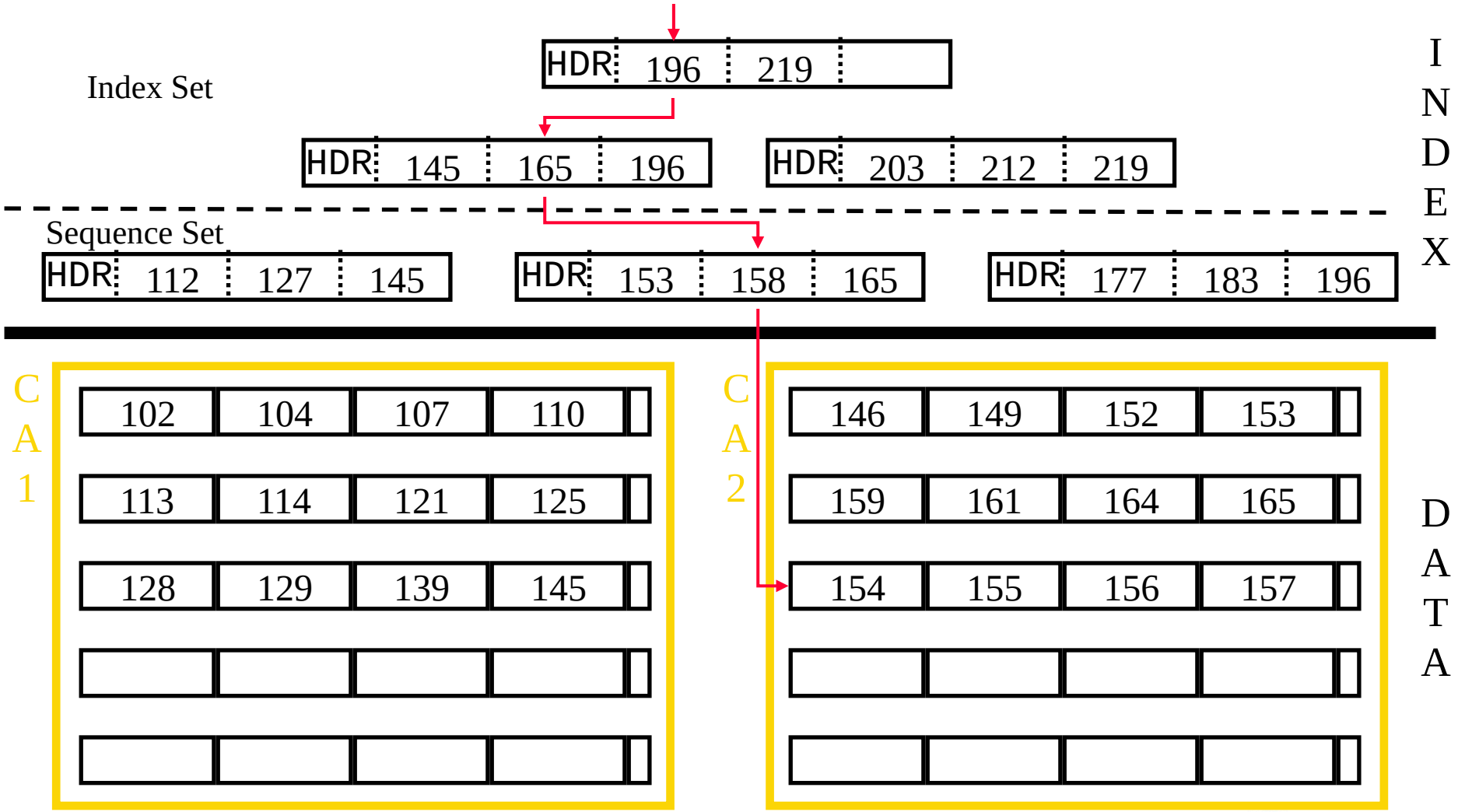
With JCL

- RECORG=KS - KSDS
- RECORG=ES - ESDS
- RECORG=RR - RRDS
- RECORG=LS - LDS

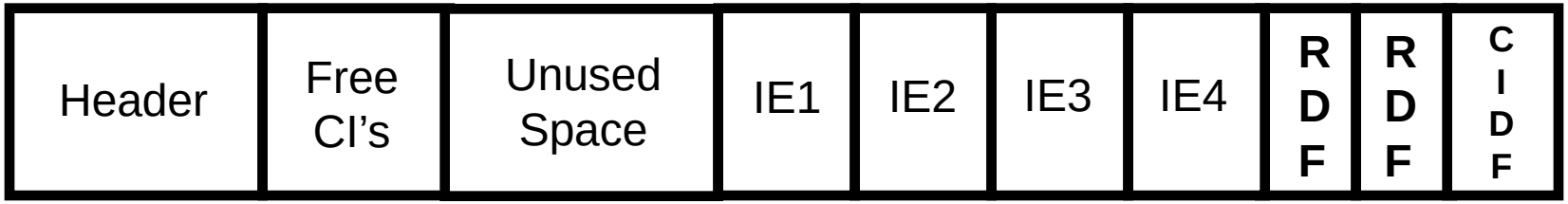
Structure of a KSDS



Locating Record with Key=156



KSDS Index Record



- Index record length is CISIZE - 7
- Header is 24 bytes long
- Index Entries contain compressed keys + 3 bytes to describe:
  - F: Number of characters eliminated from the front
  - L: Number of characters left after compression
  - P: Vertical pointer

KSDS Key Compression

- Front compression eliminates the leading characters on the key that are the same as the preceding key in the index.
- Rear compression eliminates the insignificant values from the end of the key. The current index key is compared to the next data key, and the characters to the right of the first character that is unequal to the corresponding character in the following key are eliminated.

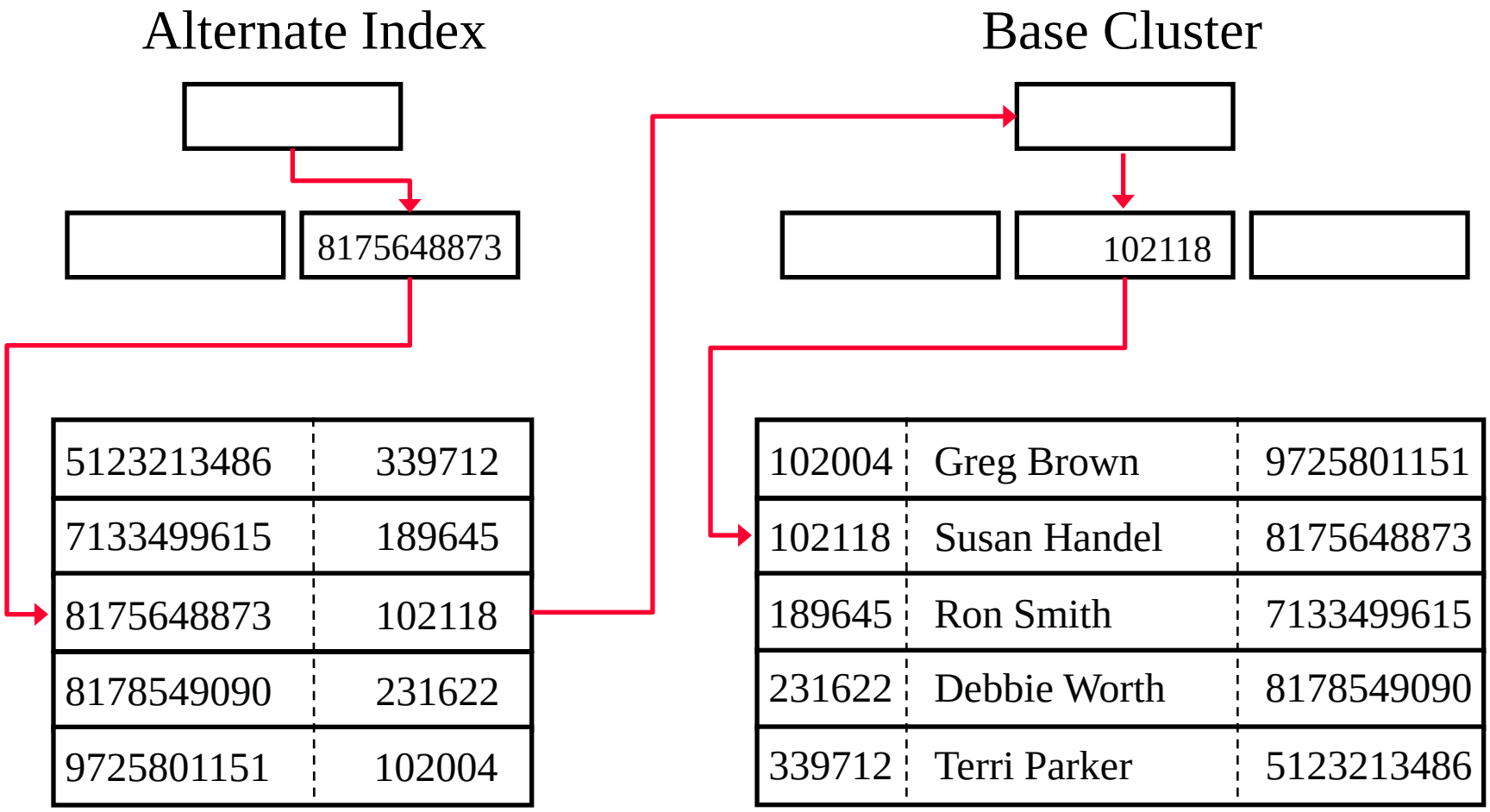
Key Compression Example

Full DATA Key	Eliminate from Front	Eliminate from Rear	Index Entry
12345	None	5	1234 0 4 0
12350			
12353			
12354			
12356	123	None	56 3 2 1
12357			
12358			
12359	1235	9	4 0 2
12370			
12373			
12380			
12385			
12390			
12401	12	None	401 2 3 3
12405			

Using an Alternate Index

- Provides another way to access data in a VSAM KSDS or an ESDS
- Allows access back to the base cluster through a PATH definition
- Supports UPGRADE so that when a record accessed through the base cluster is updated the corresponding record in the AIX is also updated

Accessing Records with an Alternate Index



Describing the Records to be Loaded

- RECORDSIZE(average, maximum)
- SPANNED or NONSPANNED
- KEYS(length, offset)

Specifying the Blocksize for a VSAM Data Set

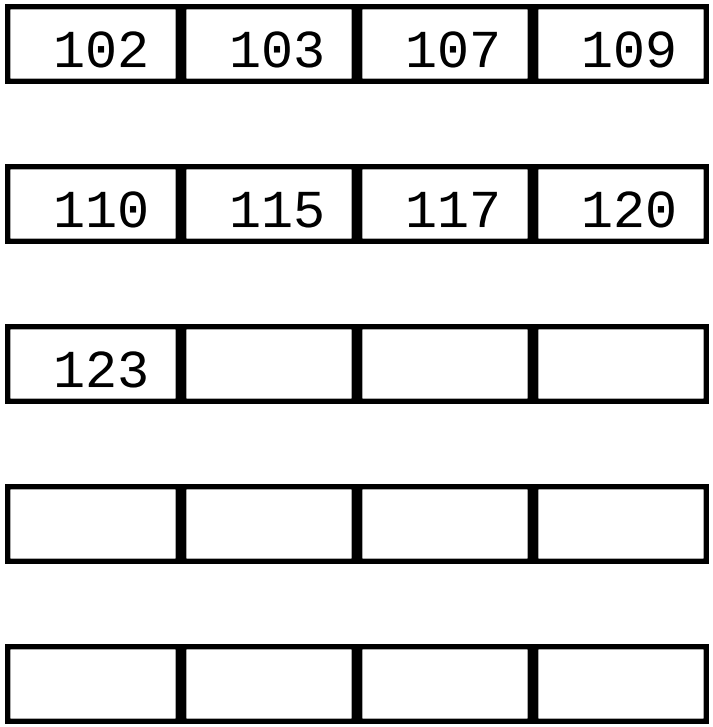
- CONTROLINTERVALSIZE(nnnn)
  - For NONSPANNED, the CI size must be at least 7 bytes larger than the max record size
  - CI size can be 512 to 8192 bytes in 512 byte increments or 8KB to 32KB in 2KB increments
  - For Linear data sets, the CI size can be 4096 to 32768 in 4096 increments
  - VSAM will adjust if a valid size is not specified
  - Usually different sizes for DATA and INDEX component
- FREESPACE(CI% CA%)

CI Size and Disk Utilization

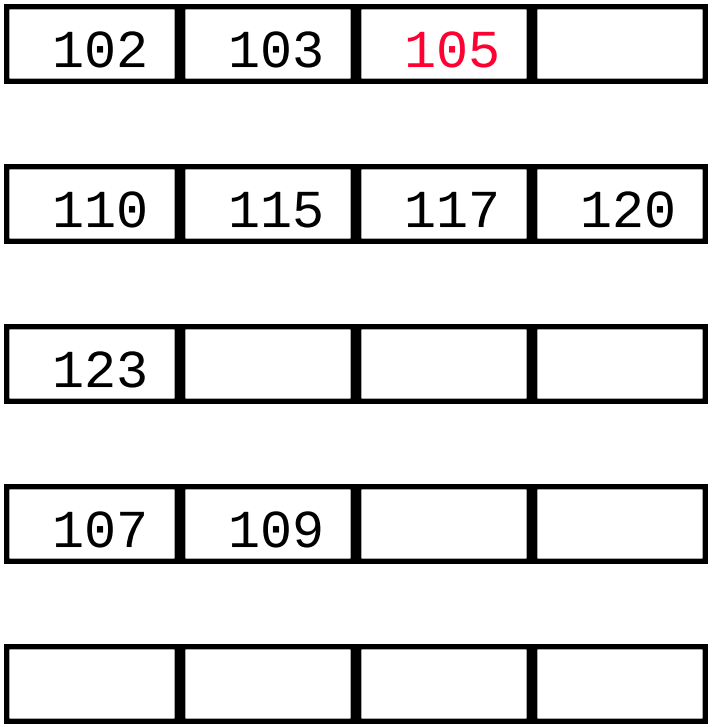
DATA CI Size	CI/CA Non-EF	Disk Utilization (%) Non-EF	CI/CA Extended Format	Disk Utilization (%)Extended Format
512	735	45	720	46
1024	495	61	495	63
1536	390	72	375	72
2048	315	78	315	81
2560	255	79	255	82
3072	225	83	225	87
3584	195	84	195	88
4096	180	89	180	92
4608	150	83	150	87
5120	135	83	135	87
5632	135	92	135	95
6144	120	89	120	92
6656	105	84	105	88
7168	105	91	105	94
7680	90	83	90	87
8192	90	89	90	92
10240	75	93	75	96
12288	60	89	60	92
14336	52	90	45	81
16384	45	89	45	92
18432	45	100	40	88
20480	37	91	37	95
22528	33	90	33	93
24576	30	89	30	92
26624	30	96	30	100
28672	26	90	22	81
30720	25	93	25	86
32768	22	87	22	92

CI Split Processing

Before Insert of Record 105



After Insert of Record 105



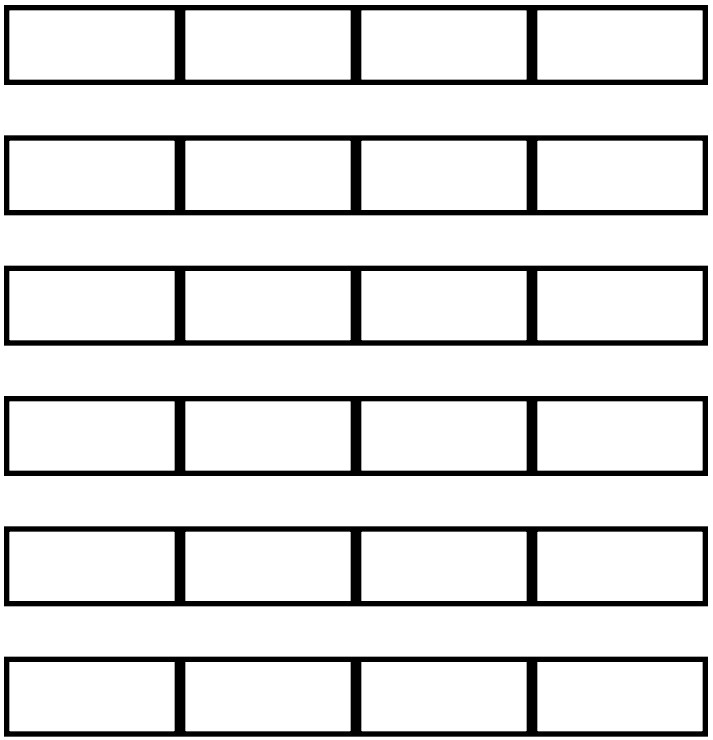
CA Split Example - Before

Insert Record 105

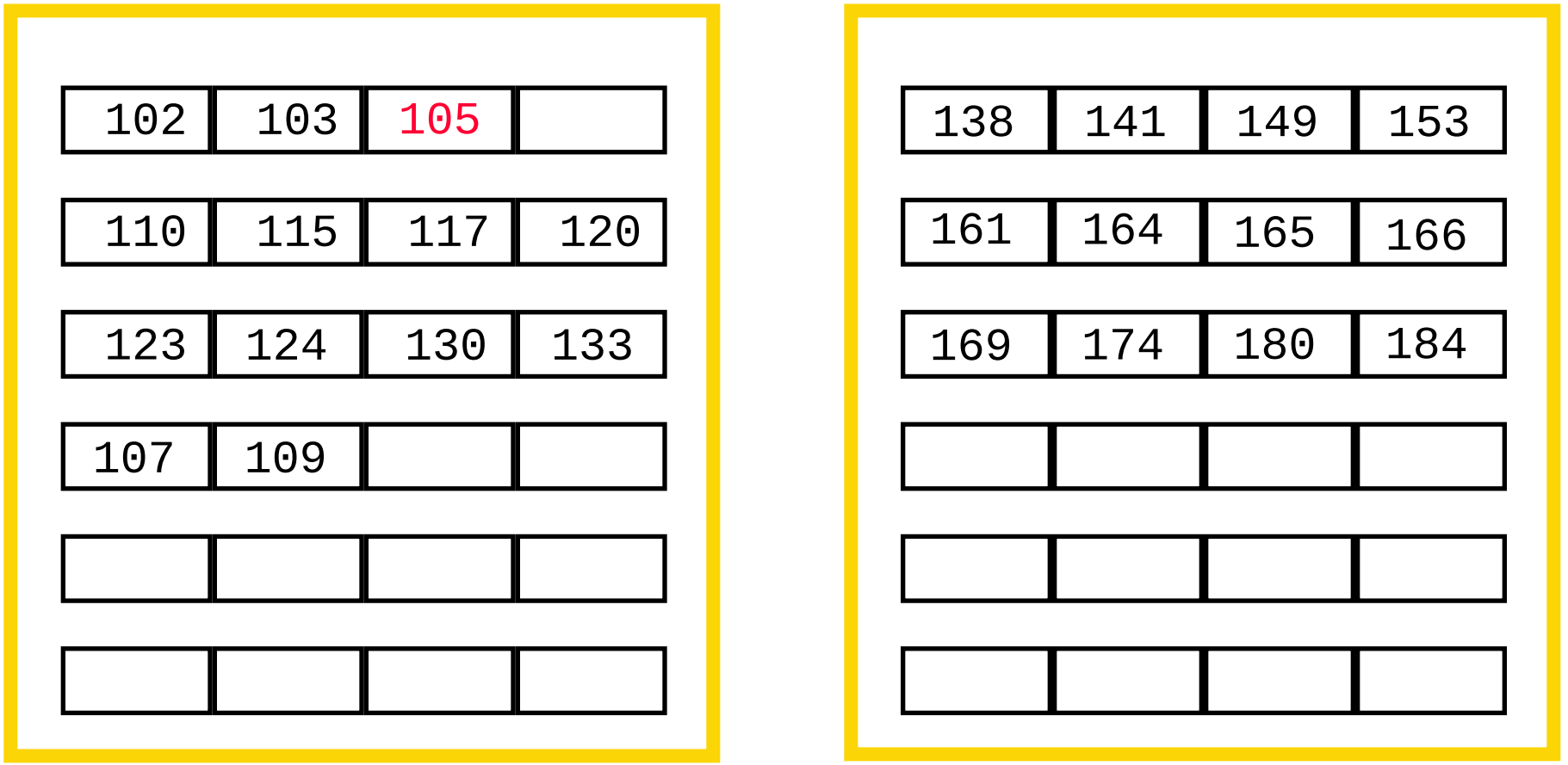
Control Area

102	103	107	109
110	115	117	120
123	124	130	133
138	141	149	153
161	164	165	166
169	174	180	184

Control Area



CA Split Example - After



CI & CA Splits Can Be Good

- The existence of CI and CA splits does not cause performance problems
- Splits create free space right in the areas of the data set where the inserts are being done
- Reorganizing a data set removes this free space and can cause the data set some performance problems until the splits can be done again
- Data sets should only be reorganized to reclaim disk space

Two Types of Split Processing

- The type of split that occurs is determined by the strategy specified at open time
  - NIS – Normal Insert Strategy
  - SIS – Sequential Insert Strategy
- Specifying the strategy in Assembler
  - MACRF=NIS|SIS
- Specifying the strategy in COBOL
  - ACCESS MODE IS RANDOM|DYNAMIC

When to Specify NIS or SIS

- ACCESS IS RANDOM should be used to when record insertion is random
- ACCESS IS DYNAMIC should be used for mass sequential insertion

Mainstar White Paper

VSAM Performance - NIS Versus SIS Mode Splits
By Ronald K. Ferguson and Elliot Hamilton

Mainstar is now part of Rocket Software.

Mass Sequential Insertion Tests

- Test #1 – ACCESS IS RANDOM and insert 30,000 records in sequential order
  - CI Splits – 2,003
  - CA Splits – 28
  - EXCP's – 32,717
  - File size grew from 34 cylinders to 62 cylinders
- Test #2 – ACCESS IS DYNAMIC and same 30,000 records as Test #1
  - CI Splits – 1
  - CA Split – 11
  - EXCP's – 3,511
  - File size grew from 34 cylinders to 44 cylinders

Random Insertion Tests

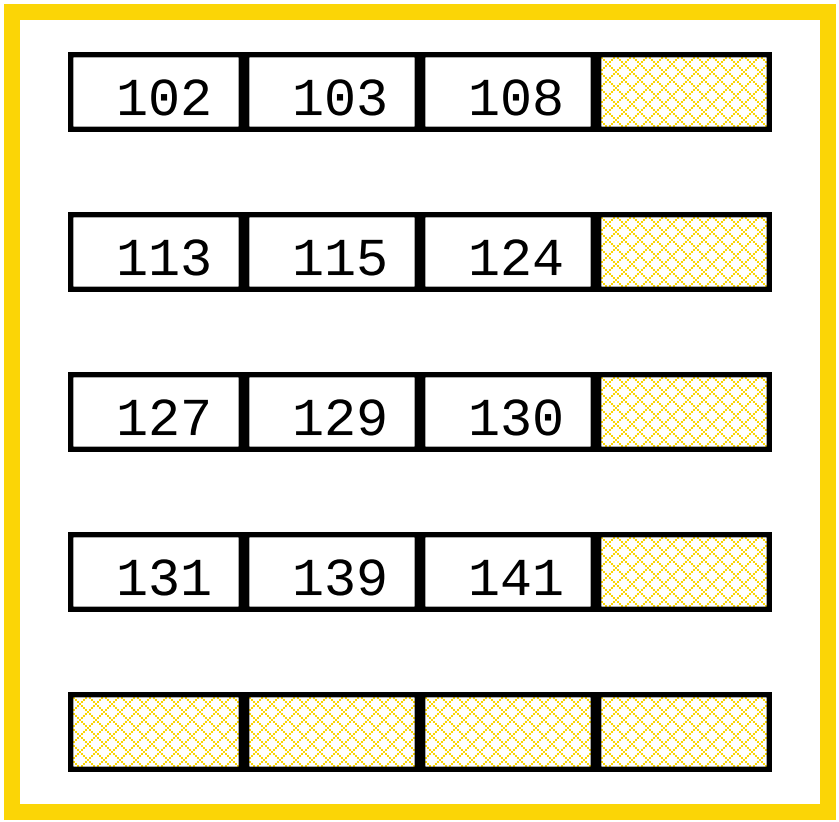
- Test #1 – ACCESS IS DYNAMIC and insert 10,000 records in random order
  - CI Splits – 4,176
  - CA Splits – 154
  - EXCP's – 40,587
  - File size grew from 34 cylinders to 187 cylinders
- Test #2 – ACCESS IS RANDOM and same 10,000 records as Test #1
  - CI Splits – 3,576
  - CA Split – 20
  - EXCP's – 20,545
  - File size grew from 34 cylinders to 54 cylinders

Some Splits Are Bad

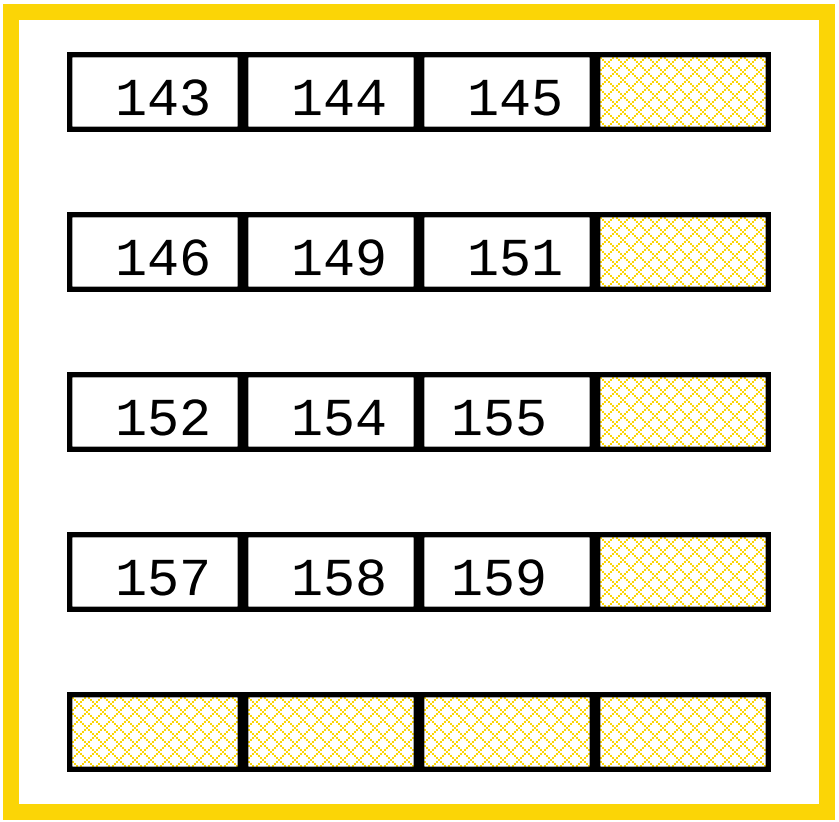
- Specifying the wrong insert strategy could cause excessive splits and increase CPU utilization and job elapse time
- Unnecessary CI and/or CA splits can increase response time for online transactions

FREESPACE(25 20)

Control Area



Control Area



FREESPACE Is Not “Free”

```
VSAM KSDS:
DATA CISZ = 4096
CI/CA = 180
FREESPACE(15,20)

Total bytes in CA = 4096 * 180 = 737,280

CA Free Space = (180 * .2) * 4096 = 147,456

CI Free Space = 4096 * .15 * (180 * .8) = 88,560

Percentage of each CA reserved for Free Space =
      (147,456 + 88,560) / 737,280 = 32%
```

VSAM Features for Extended Format Only

- Compression
- VSAM partial space release
- Enhanced multi-volume allocation support
- Extended Addressability
- Striping
- System-managed Buffering (SMB)

SHAREOPTIONS Parameter

- SHAREOPTIONS(n x)
  - “n” is the option for cross-region sharing requirements
  - “x” is the option for cross-system sharing requirements
- SHAREOPTIONS is ignored for VSAM data sets opened for Record Level Sharing (RLS)

Cross-region Sharing

- 1 - any number of users for READ OR one user for UPDATE; VSAM ensures complete integrity
- 2 - any number of users for READ AND one user for UPDATE; VSAM ensures only write integrity
- 3 - the data set can be fully shared, but it is the user's responsibility to maintain integrity
- 4 - same as option 3, but VSAM will refresh buffers for direct requests

Cross-system Sharing

- 3 - the data set can be fully shared, but the user is responsible for maintaining the integrity of the data set
- 4 - the same as option 3, but VSAM will refresh the buffers for each direct processing request

SPEED vs. RECOVERY

- RECOVERY tells VSAM to write a data record and then an EOF record at load time
- SPEED tells VSAM not to pre-format the data component's space at load time
- RECOVERY is the default, so always specify SPEED

Types of VSAM Buffering

- Non-Shared Resources (NSR)
- Local Shared Resources (LSR)
- Global Share Resources (GSR)
- Record Level Sharing (RLS)

System Managed Buffering

- Replaces Batch LSR (BLSR)
- Is only available for Extended Format data sets
- Can be implemented via a DATACLAS definition or using the AMP JCL parameter
- Introduced new JCL keywords to support SMB
  - AMP='ACCBIAS=USER|SYSTEM|DO|DW|SO|SW'
  - AMP='SMBDFR=Y|N'
  - AMP='SMBHWT=nn'
  - AMP=SMBVSP=nnK|M'

Optimization Technique Selected

	SEQ BIAS in STORCLAS	DIR BIAS in STORCLAS	SEQ & DIR BIAS in STORCLAS	No BIAS Specified in STORCLAS
MACRF=SEQ (this is the default)	**SO**	**SW**	**SO**	**SO**
MACRF=(SEQ,SKP)	**SO**	**SW**	**SW**	**SW**
MACRF=DIR	**DW**	**DO**	**DO**	**DO**
MACRF=(DIR,SEQ) or MACRF=(DIR,SKP) or MACRF=(DIR,SEQ,SKP)	**SW**	**DW**	**DW**	**DW**

SMB Load Mode Options

- Optimization when the HURBA=0 and the data set is opened in LOAD mode
  - Create Optimize (CO)
  - Create Optimize Recovery (CR)
- These options cannot be specified by the user

Additional Tuning Options When ACCBIAS=DO

- SMBDFR=Y|N
  - Default for SHR(1,3) & SHR(2,3) is SMBDFR=Y
  - Default for all other share options is SMBDFR=N
- SMBVSP=nK|nM
- SMBHWT=n
  - Default is SMBHWT=0
  - The “n” is a weighted value between 0 - 99

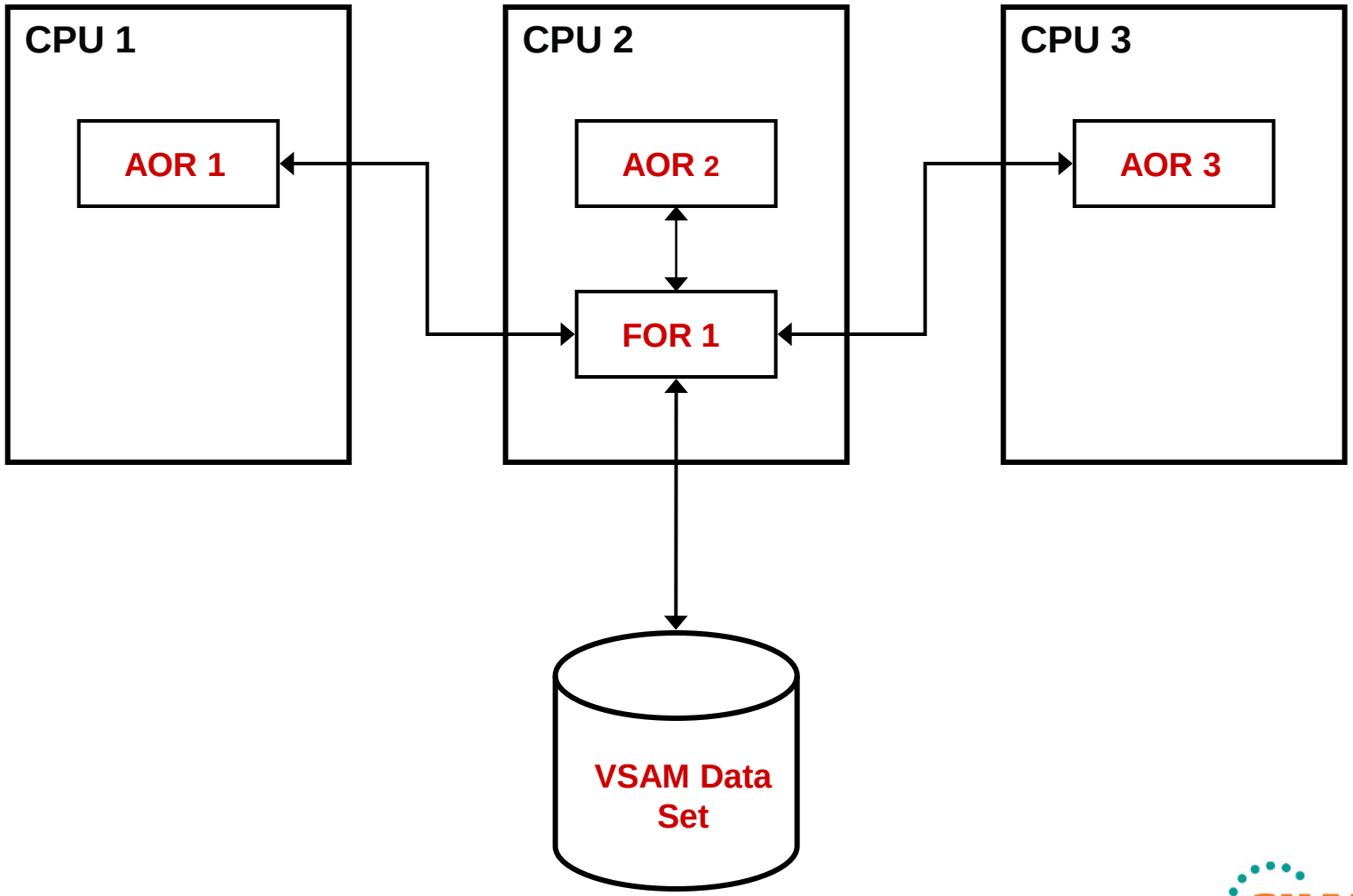
Controlling Where the SMB Buffers Get Loaded

- AMP='RMODE31'
  - ALL
  - BUFF – Buffers only
  - CB – Control blocks only
  - NONE

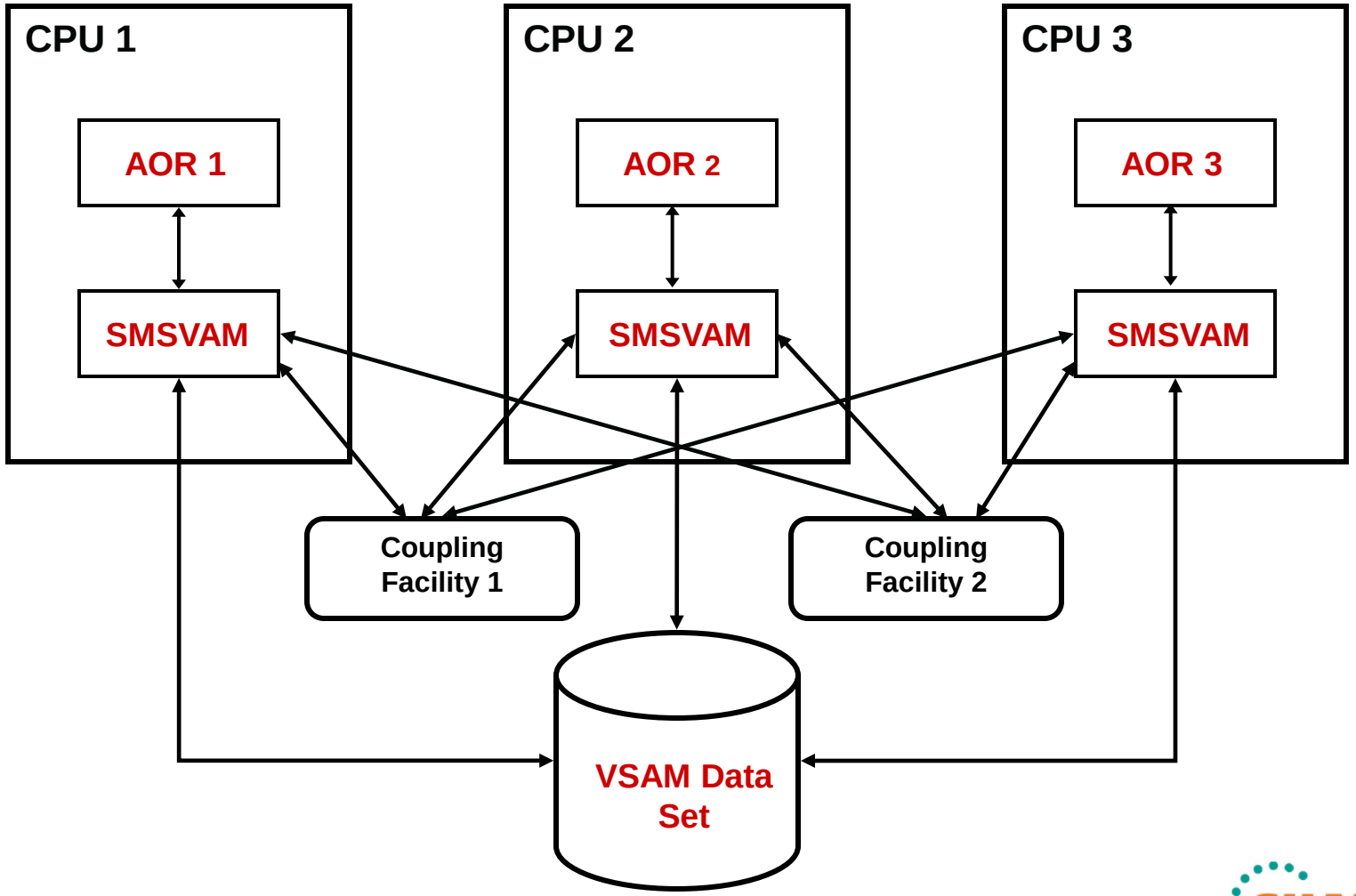
VSAM Record Level Sharing

VSAM Record Level Sharing is a function that allows VSAM data sets to be fully shared with data integrity among multiple user across multiple systems.

CICS VSAM Sharing Today



CICS Sharing with RLS



Advantages of Using VSAM RLS

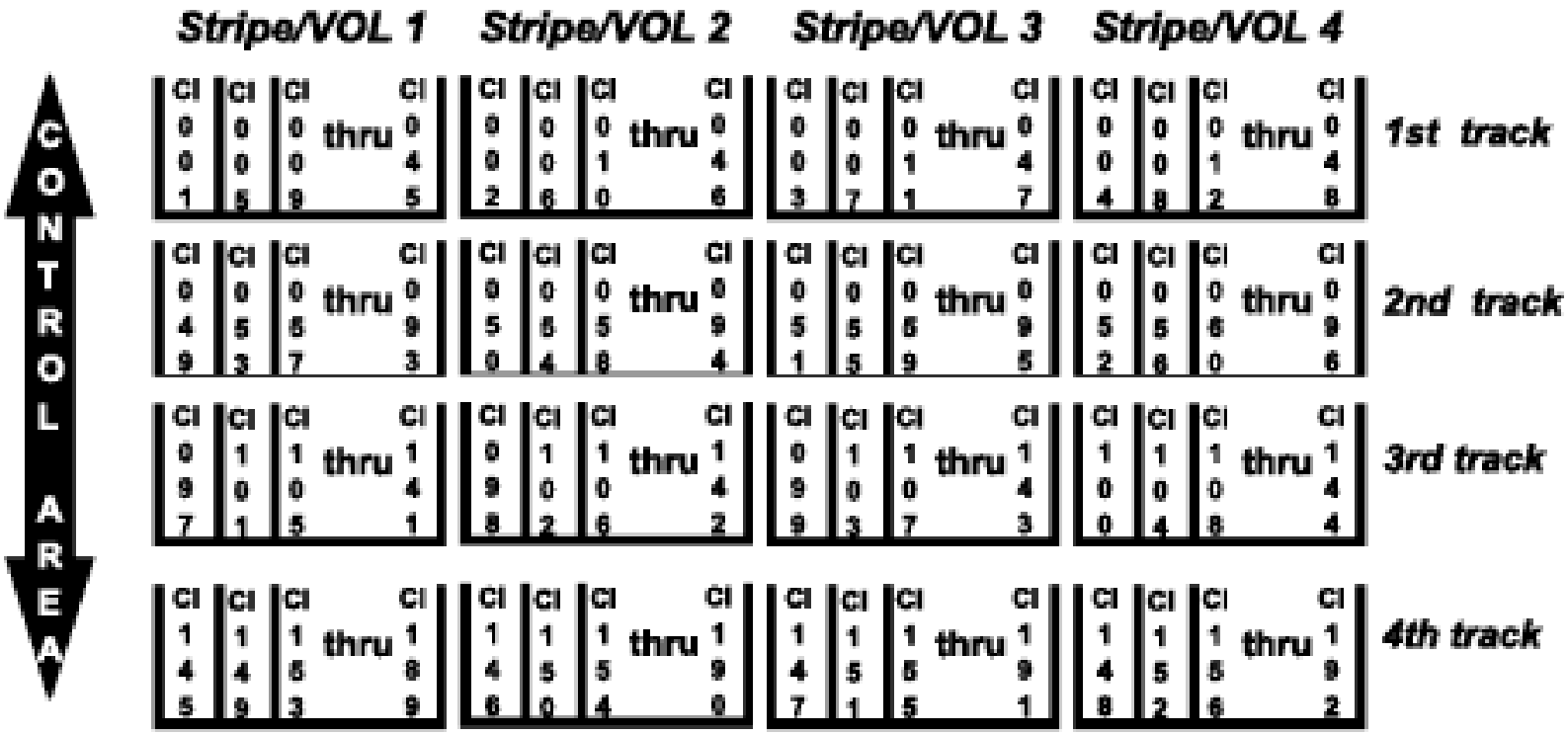
- Increases data availability
  - No Single Point of Failure (SPOF)
  - Data remains available during both planned and unplanned outages
- Improves data integrity
  - No more “dirty reads” if another application is updating the data set
  - All users are aware of the HARBA and HURBA
  - Locks are held in the event of a CICS failure
- Eliminates the processor constraint for a single FOR
- Provides flexibility in balancing workloads
- Provides the base for Transactional VSAM

VSAM Striping

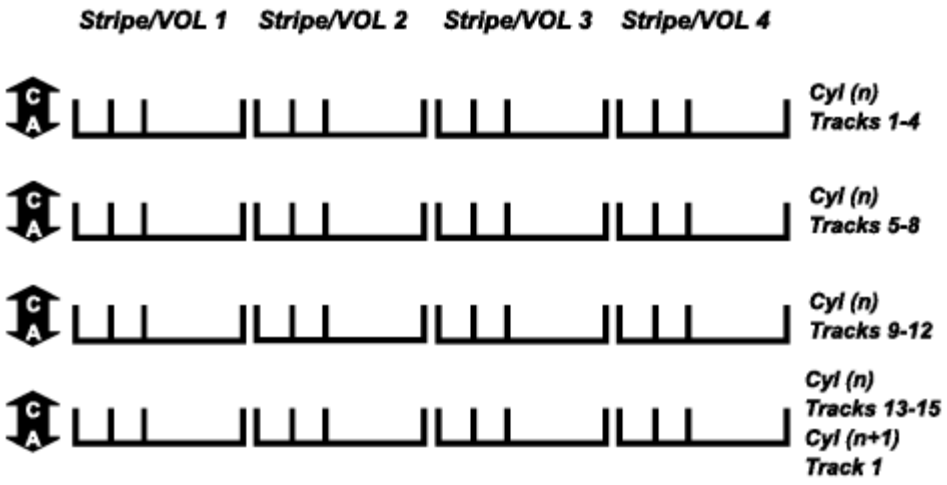
- Improves throughput for sequential, I/O bound applications
- Spreads control intervals in a control area across multiple devices
- Does not adversely impact the performance of random I/O applications
- More effectively performs the same function that VSAM KEYRANGE data sets did

VSAM Striping Layout

Data CI size - 4k, Physical Blocksize - 4K
4K blocks per 3390 track - 12, Stripe count - 4
CYL (n n)

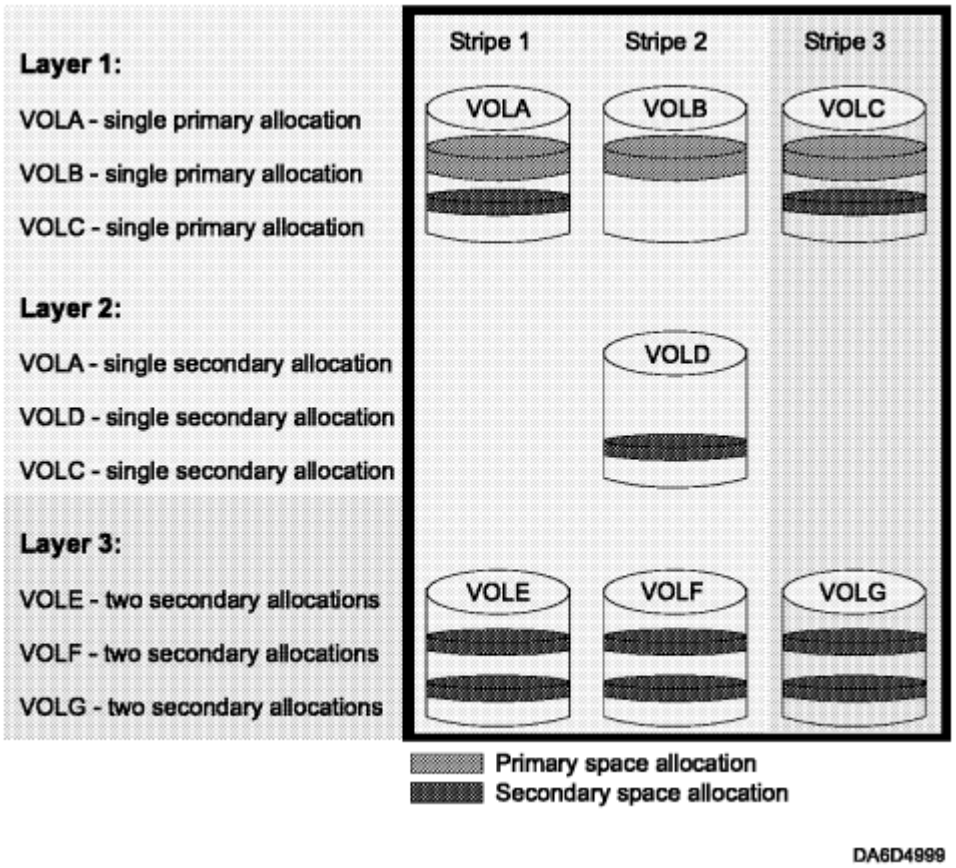


CI/CA Layout for Striped Data Sets



- Control Area (CA) size = 16 tracks spread across the four stripes
  - Cylinder allocation specified
  - Increased Index CL size requirement

Layering for Striped Data Sets



Using IDCAMS LISTCAT

- Syntax ==>> LISTCAT ENTRY(entry) ALL
- Provides extensive information about VSAM data sets including:
  - Data set attributes
  - SMS information
  - RLS information
  - Allocation information (like primary and secondary space requested, CI size, split information, etc.)
  - Volume and extent information
  - Sphere information

Additional VSAM Information

- DFSMS Using Data Sets Manual
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/DGT2D4A0/CCONTENTS?SHELF=ez2zo111&DN=SC26-7410-11&DT=20110606092005
- VSAM Demystified
  - http://www.redbooks.ibm.com/redpieces/abstracts/sg246105.html?Open