VSE/VSAM

Session E45

VSE/VSAM Concepts and Tuning Tips

Dan Janda
VM/ESA and VSE/ESA Systems Center
Endicott, New York
janda@us.ibm.com

VSE/VSAM

Abstract

The objectives of this presentation are to provide an overview of the VSE/VSAM Subsystem's features and how to exploit them for best system performance characteristics.

The presentation focuses on facilities available with the levels of VSE/VSAM available with VSE/ESA, particularly since VSE/ESA 1.3's availability. VSE/ESA 2.1 and later changes are highlighted.

Changes of VM and VSE operating modes and releases impact performance and can impact customer throughput, response time and satisfaction. Highlights and "gotchas" in the performance area are discussed.

The author has worked with VSAM for over 25 years, and has extensive experience optimizing the performance of IBM VSE and VM/VSE computer systems using VSAM file systems.

Bibliography

– **For more information please see the following publications;**

VSE/ESA 1.3 and 1.4 level:

Using VSE/VSAM Commands and Macros	GC33-6532
VSE/VSAM User's Guide	GC33-6535

VSE/ESA 2.1 and 2.2 level:

VSE/VSAM Commands	GC33-6631
VSE/VSAM User's Guide and Application Programming	GC33-6632

VSE/ESA 2.3 level: The above (2.1/2.2) manuals plus

VSE/ESA Enhancements Ver 2 Rel 3 or	SC33-6629(-01)

VSE/ESA 2.4 level:

VSE/VSAM Commands (Version 6 Release 4)	SC33-6731
VSE/VSAM User's Guide and Application Programming	SC33-6732

The VSE/ESA World Wide Web Home Page: http://www.s390.ibm.com/vse/

Other Information, Trademarks, etc.

You may also be interested in Bill Merrow's book, ***VSE/ESA Performance Management and Fine Tuning***, published by McGraw-Hill, 1993. (ISBN 0-07-041753-9) Bill's discussion of VSAM topics are still useful.

The following words used in this presentation are trademarks or registered trademarks of the IBM Corporation.

VSE   VSE/ESA   ESA   VSE/VSA   S/390

VSE/VSAM Overview

Disk Storage devices require complex management for good performance and utilization

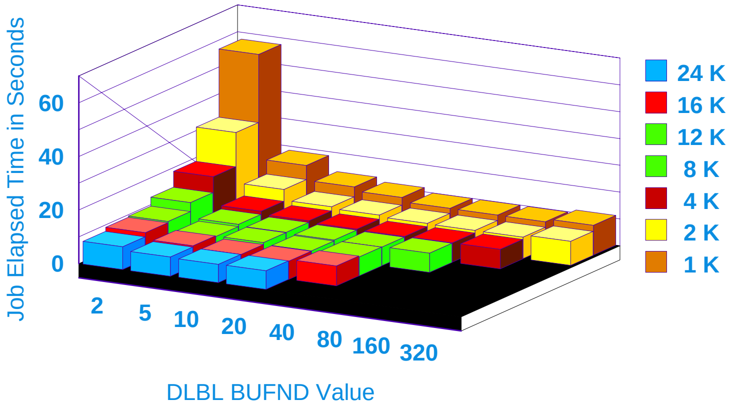
- **Physical Organization of disks**
  - **Devices**
  - **Cylinders**
  - **Tracks**
  - **Records**
  - **Control Information**
- **Control information includes**
  - **Data verification and correction information**
  - **Directory information**

VSE/VSAM Performance3

VSE/VSAM provides opportunities to greatly affect

- **Online response time, Batch elapsed time...**

File Load Time Variation
vs. CI Size and BUFND

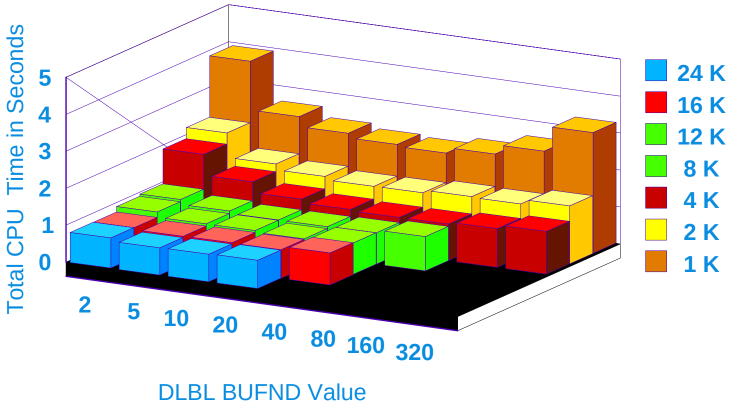


VSE/VSAM Performance

VSE/VSAM provides opportunities to greatly affect

- **CPU and I/O Resource usage**

Total CPU Time Variation
vs. CI Size and BUFND



Access Methods

Access Methods are a combination of defined architectures, protocols, and program instructions designed to assist users in managing system resources, such as disk drives

VSE/ESA Access Methods include:

- **SAM for Disk, Tape, and Unit Record devices [Sequential Access Method]**
- **BDAM for Disk [Basic Direct Access Method]**
- **VTAM (for communications) [Virtual Telecommunications Access Method]**
- **VSAM (for Disk) [Virtual Storage Access Method]**

VSE/VSAM

Virtual Storage Access Method

General Purpose Disk Access Method

- **Data Set (file) Organizations Supported**
  - **Entry Sequenced (ESDS)**
    - **Records stored in the order presented**
  - **Key Sequenced (KSDS)**
    - **Records stored in key sequence**
  - **Alternate Index (AIX)**
    - **Provision for multiple search keys**
  - **Relative Record (RRDS)**
    - **Records stored in numbered, fixed length slots**
  - **Variable Length Relative Record (VRDS)**
    - **Records stored in numbered, variable length slots**
  - **SAM ESDS (SAM-ESDS)**
    - **Logical records stored in SAM blocks within VSAM CIs**

VSE/VSAM

10Virtual Storage Access Method

General Purpose Disk Access Method...

- **Data Set (file) Access Modes Supported**
  - **Sequential**
    - **Records presented in the order stored**
  - **Direct**
    - **Records presented in arbitrary order**
  - **Indexed**
    - **Records presented in an index's order**
  - **Keyed**
    - **Records identified by key field contents**
  - **Addressed**
    - **Records identified by their relative address within data set**
  - **Sequential (forward or backward)**
    - **Records identified by position within data set**

VSE/VSAM Provided Functions

Catalog

- **Volume Information**
  - **Device characteristics**
  - **Allocated pool(s) of space**
  - **Space used**
- **File Information**
  - **Record characteristics**
  - **Usage statistics**
  - **Space allocations**

VSE/VSAM Provided Functions...

DASD Space Management

- **Primary and secondary allocation**
  - **Space classes**

Performance

- **Sequential processing**
  - **Large data transfers**
- **Direct processing**
  - **Buffer look-aside**

VSE/VSAM Provided Functions...

Integrity

- **Backup/Restore**
  - **To tape and to disk**
- **Sharing controls**
  - **Intra-System**
  - **Inter-System**
  - **VSE Lock File Implementation**
    - **VM/ESA Virtual Disk for lock sharing among VSE guests of a single VM/ESA image**
    - **No lock sharing with VM/CMS**
    - **No lock sharing with OS/390**

VSE/VSAM Provided Functions...

Large File Space and Compression

- **VSE/VSAM Data Compression (VSE/ESA 2.1...)**
  - **Exploits Hardware Feature if available**
  - **Uses compatible software routine if not**
  - **Transparent to application**
- **KSDS support for data sets larger than 4 GB**
  - **Extra Large (XL) support is transparent to application software**
  - **Up to 492 GB file size, depending on DASD type and extent allocations**

VSE/VSAM Provided Functions...

Other Capabilities

- **Multiple keys**
  - **For searching and sequencing**
- **Generic searches**
  - **With partial keys**
- **Sequential access**
  - **Both forward and backward directions**

VSE/VSAM Data Organizations

Components

Data	Alternate Index	Volume
Index	Path	Data Space

Clusters

- **All the components of a data set together**

Spheres

- **All the components, including all related Alternate Index data sets and paths**

VSE/VSAM

VSE/VSAM Data Organizations...

KSDS -- Key Sequenced Data Set

- Stored logically in key sequence
- Indexed
- Data can be compressed
- Larger than 4 GB data set

ESDS -- Entry Sequenced Data Set

- Stored in the order added
- Non-indexed
- Data can be compressed

VSE/VSAM

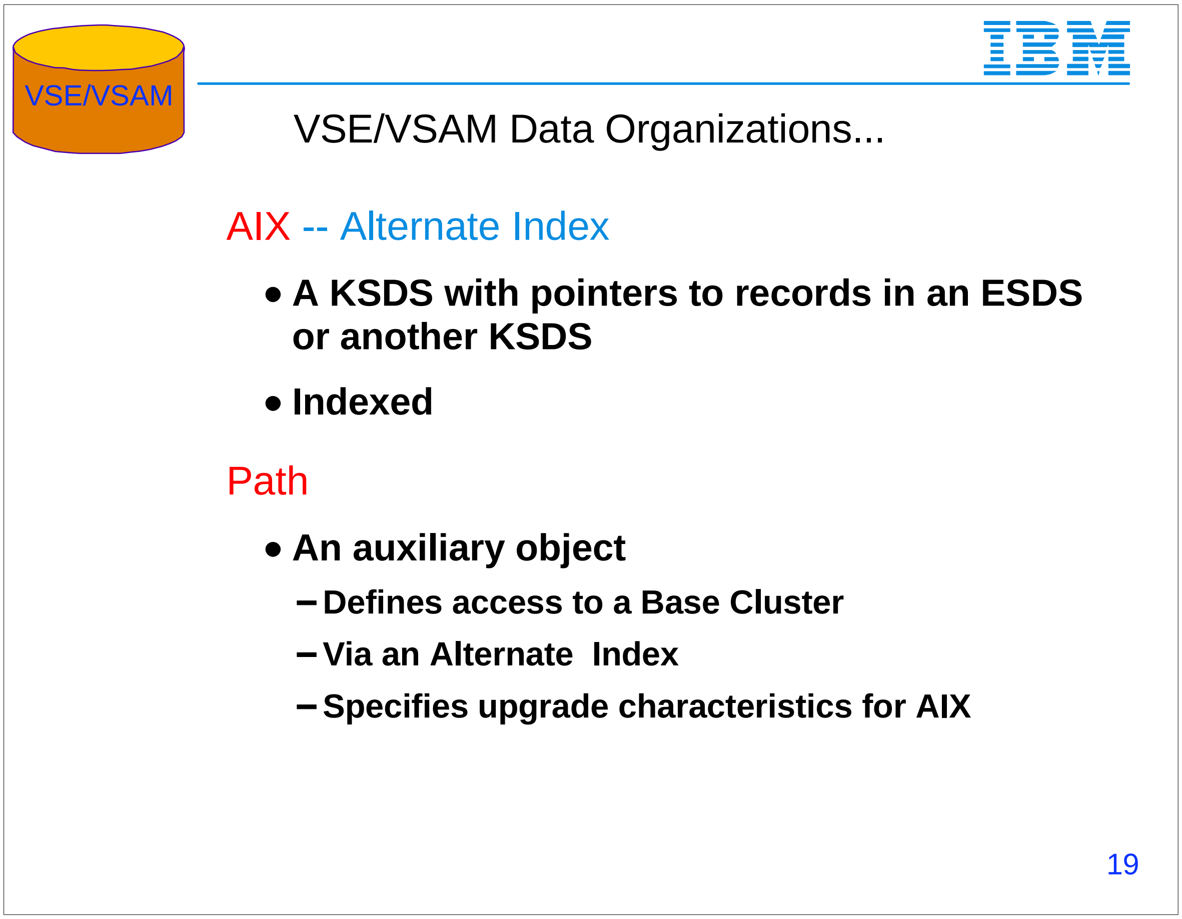
VSE/VSAM Data Organizations...

RRDS -- Relative Record Data Set

- Stored in relative record order
- Non-indexed
- Fixed length records only
- Data cannot be compressed

VRDS -- Variable Relative Record Data Set

- Stored logically in relative record order
- Indexed
- Data can be compressed



VSE/VSAM

IBM

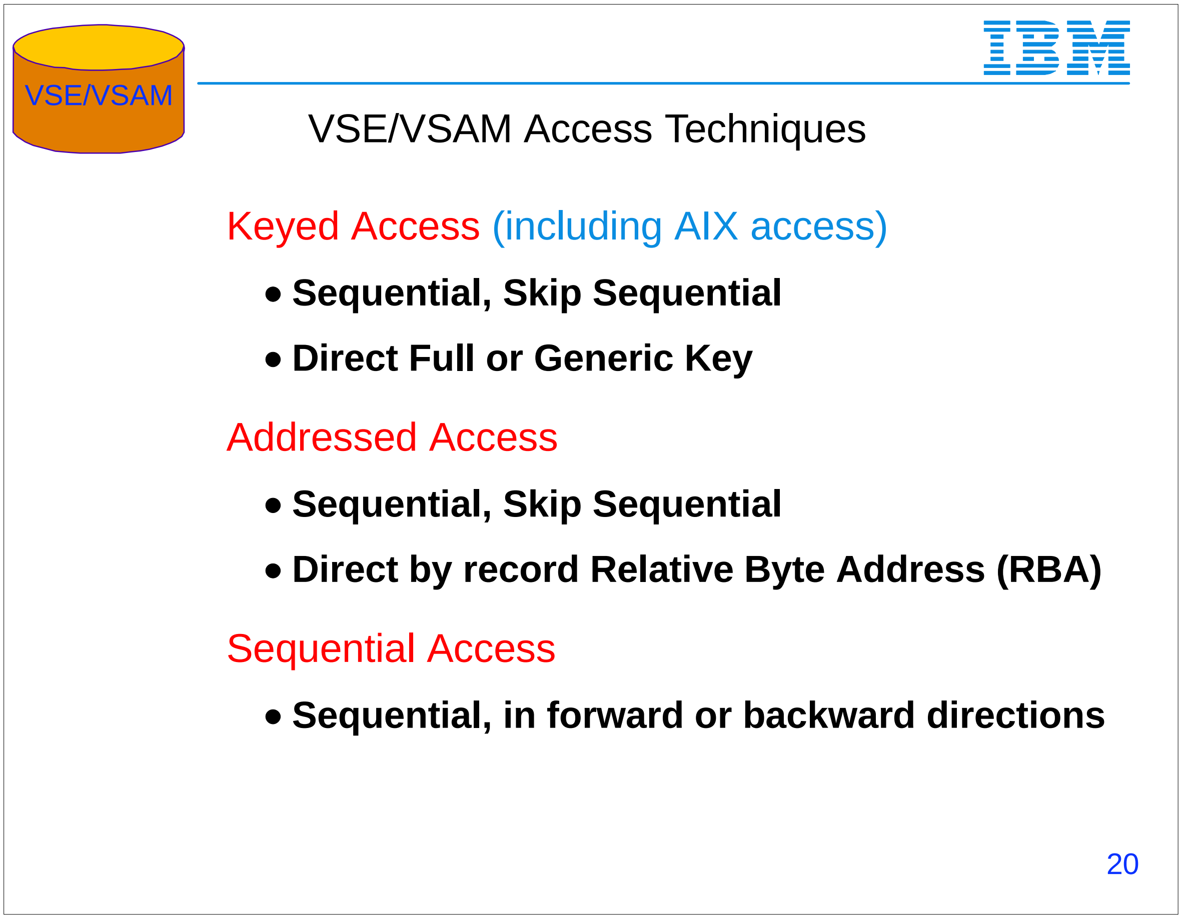
VSE/VSAM Data Organizations...

AIX -- Alternate Index

- **A KSDS with pointers to records in an ESDS or another KSDS**
- **Indexed**

Path

- **An auxiliary object**
  - **Defines access to a Base Cluster**
  - **Via an Alternate Index**
  - **Specifies upgrade characteristics for AIX**



VSE/VSAM

IBM

VSE/VSAM Access Techniques

Keyed Access (including AIX access)

- **Sequential, Skip Sequential**
- **Direct Full or Generic Key**

Addressed Access

- **Sequential, Skip Sequential**
- **Direct by record Relative Byte Address (RBA)**

Sequential Access

- **Sequential, in forward or backward directions**

VSE/VSAM Physical Data Storage

Logical Record

- Group of related data elements or fields
- Logically stored and retrieved together
- May have a key or record number
- Fixed or variable length
- Can be compressed when stored and expanded when retrieved
- Can span one or more Control Intervals up to the length of a Control Area

VSE/VSAM Physical Data Storage...

Logical Record...

VSE/VSAM Logical Record

Data-1	Key	Data-2	*A-Key-1*	...

Data-1 0 to n bytes data (optional). Not compressed

Key Logical record key, not present for ESDS, RRDS, VRDS. Not compressed

Data-2 0 to n bytes data. Compressible if 40 bytes or greater (may include alternate key fields)

A-Key-1 An alternate key (optional) Multiples permitted, Compressible if in Data-2 area

Alternate keys may be located anywhere in the logical record, including overlapping the primary key.

Compression ratios vary depending on record data content

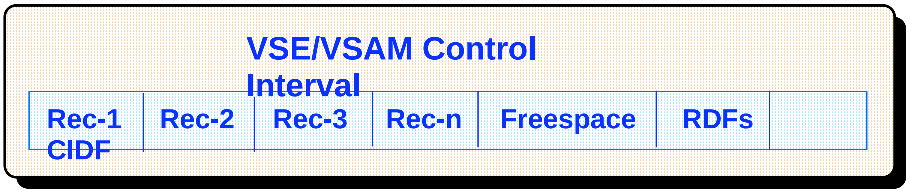
VSE/VSAM Physical Data Storage...

Control Interval (CI)

- **The smallest unit of data transfer between disk and main storage buffers**
- **One or more logical records (or logical record segments, in the case of spanned records)**
- **One or more physical blocks on disk are used to store a CI**
- **CIs can span disk tracks, but do not span CA boundaries**

VSE/VSAM Physical Data Storage...

Control Interval (CI)...



Rec-1..Rec-n	1 to n logical records of any length
Freespace	Unused space available for increasing length of or adding logical records
RDFs	Record Descriptor Fields -- 3 bytes each for KSDS/ESDS, one per unique length or repetition, for RRDS, one per slot For fixed length records in KSDS/ESDS there will be 2 RDFs.
CIDF	Control Interval Definition Field, 4 bytes, identifies available free space

VSE/VSAM Physical Data Storage...

Control Interval (CI)...

- **CIs are addressed by Relative Byte Address, a 4-byte address relative to the beginning of the data set.**
- **Records may also be referenced by RBA (Addressed Access).**
- **The largest RBA possible is 2^32-1, about 4 billion bytes or 4 GB. This limits the maximum size of a VSAM data set.**
- **Larger KSDS are permitted by "Extra Large Data Set" or XXL support beginning in VSE/ESA 2.3. CI numbers are used instead of RBAs, permitting many times more space within a data set**
- **VSAM Data Compression is able to store more records in a given address range, depending on the achieved compression factor. This can relieve the 4 GB limit to some degree, beginning in VSE/ESA 2.1.**

VSE/VSAM Physical Data Storage...

Control Interval (CI) Recommendations

- **Generally, larger DATA CI sizes give better sequential processing performance, and for KSDS, permit smaller INDEX CI sizes as well.**
- **This can permit much or all of the CIs within an index to be held in buffers in memory, improving direct processing performance.**
- **Larger DATA CI sizes reduce the number of physical records containing data, and the length of channel programs used to read or write data from and to disk, but require additional virtual storage.**
- **Tradeoffs have changed -- virtual and real storage are more available and less expensive, and buffer search algorithms are being changed to improve their speed.**

VSE/VSAM

VSE/VSAM Physical Data Storage...

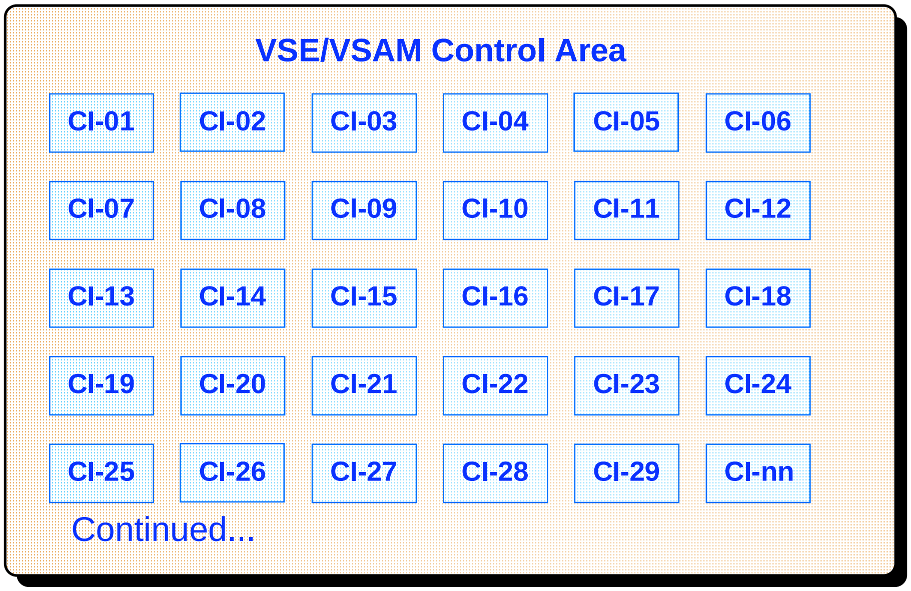
Control Area (CA)

- **A group of control intervals**
- **In a KSDS, all CIs in a CA are mapped by a single index control interval, called a "sequence set CI"**
- **There can be one or more CIs in a CA.**
- **The size of a CA is the smallest of:**
  - **A physical disk cylinder (or MAX-CA for FBA)**
  - **Primary allocation amount**
  - **Secondary allocation amount**
- **Larger CAs are generally better**

VSE/VSAM

VSE/VSAM Physical Data Storage...

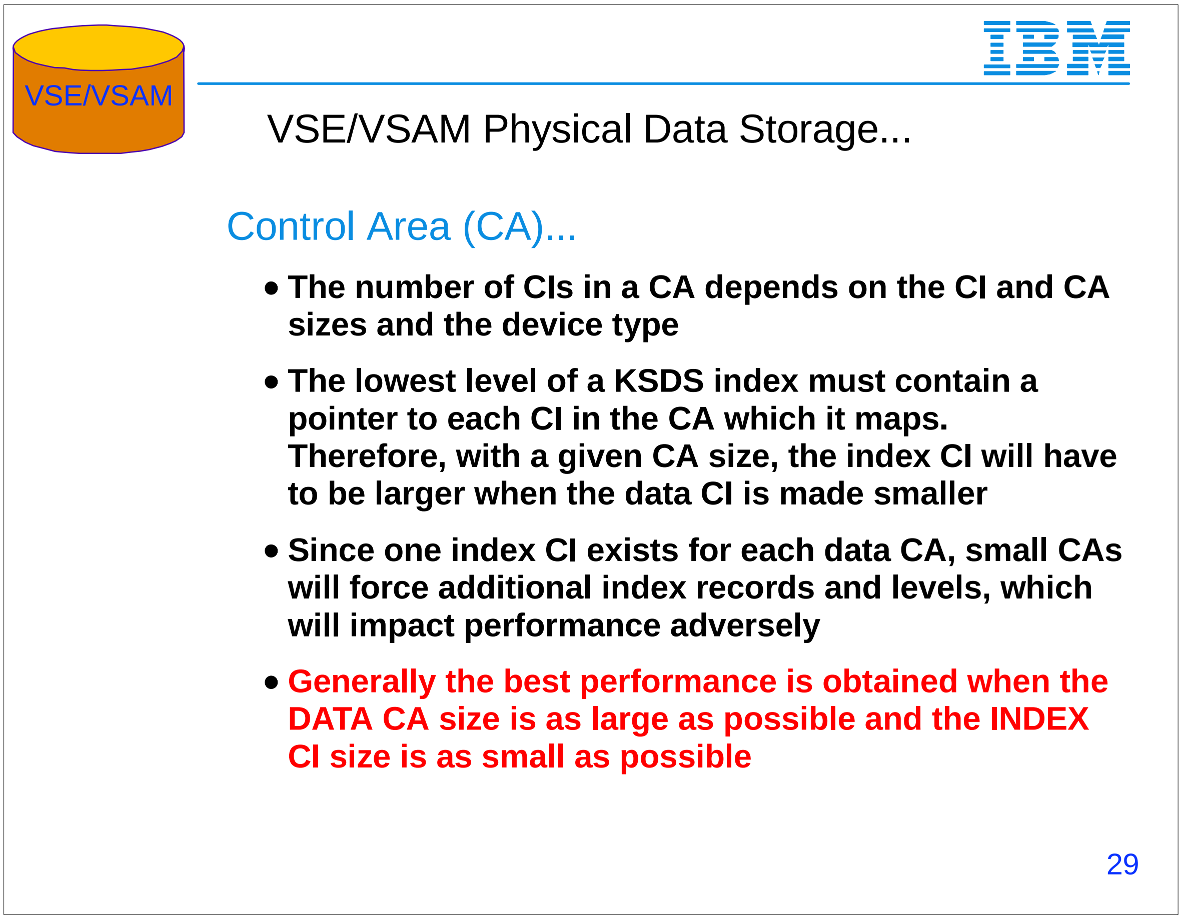
Control Area (CA)...



VSE/VSAM Control Area

CI-01	CI-02	CI-03	CI-04	CI-05	CI-06
CI-07	CI-08	CI-09	CI-10	CI-11	CI-12
CI-13	CI-14	CI-15	CI-16	CI-17	CI-18
CI-19	CI-20	CI-21	CI-22	CI-23	CI-24
CI-25	CI-26	CI-27	CI-28	CI-29	CI-nn

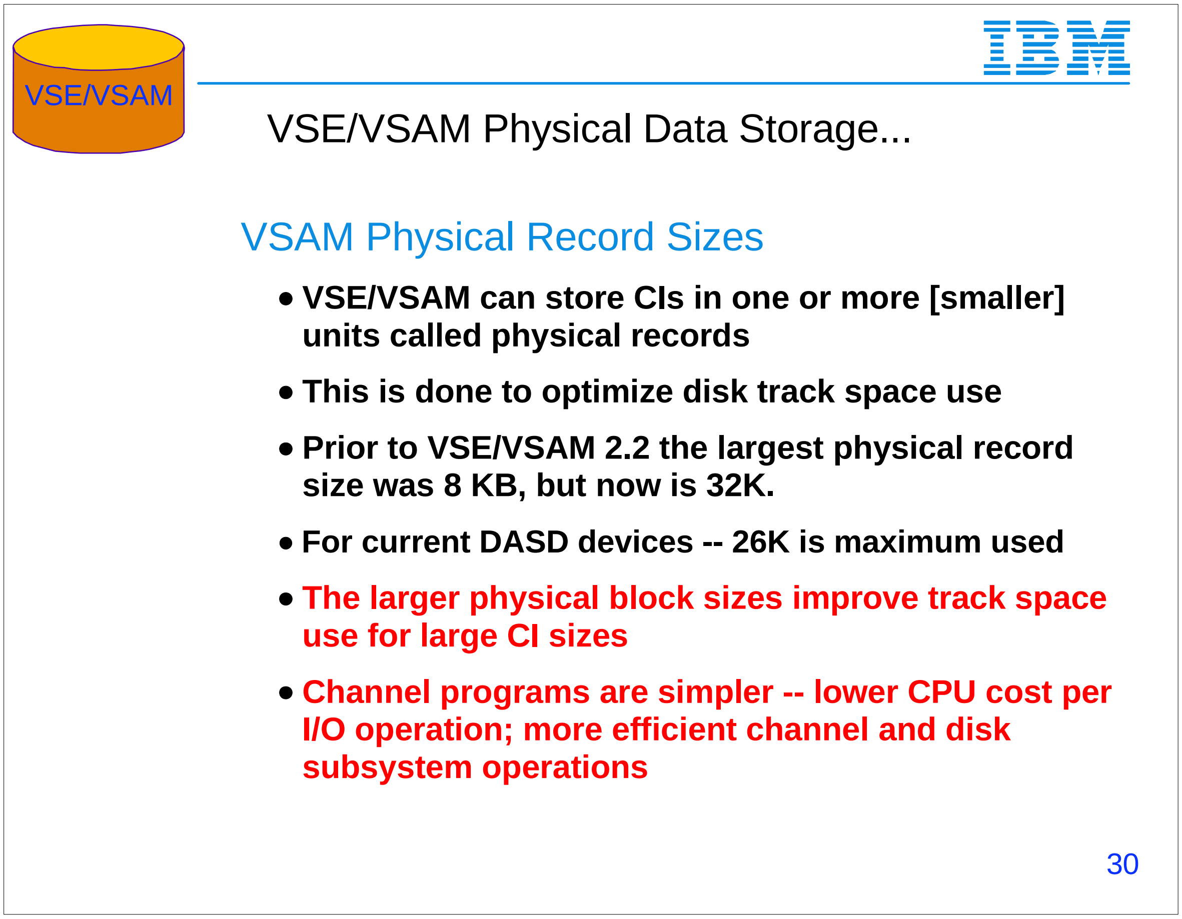
Continued...



VSE/VSAM Physical Data Storage...

Control Area (CA)...

- The number of CIs in a CA depends on the CI and CA sizes and the device type
- The lowest level of a KSDS index must contain a pointer to each CI in the CA which it maps. Therefore, with a given CA size, the index CI will have to be larger when the data CI is made smaller
- Since one index CI exists for each data CA, small CAs will force additional index records and levels, which will impact performance adversely
- Generally the best performance is obtained when the DATA CA size is as large as possible and the INDEX CI size is as small as possible



VSE/VSAM Physical Data Storage...

VSAM Physical Record Sizes

- VSE/VSAM can store CIs in one or more [smaller] units called physical records
- This is done to optimize disk track space use
- Prior to VSE/VSAM 2.2 the largest physical record size was 8 KB, but now is 32K.
- For current DASD devices -- 26K is maximum used
- The larger physical block sizes improve track space use for large CI sizes
- Channel programs are simpler -- lower CPU cost per I/O operation; more efficient channel and disk subsystem operations

VSE/VSAM

VSE/VSAM Physical Data Storage...

VSAM Physical Record Sizes...

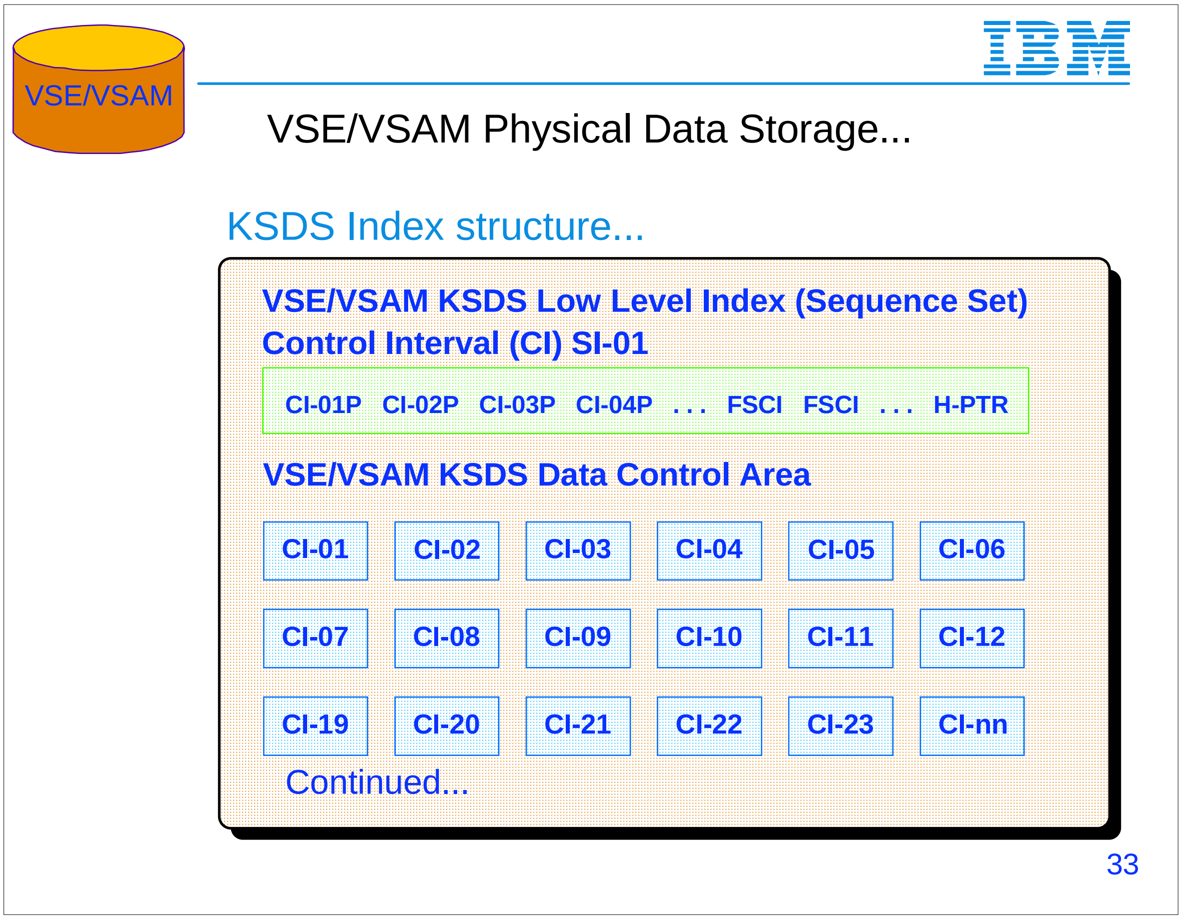
- **For choosing a CI size to maximize track use, refer to the *VSE/VSAM User's Guide and Application Programming* manual.**
- **For optimization, you must consider:**
  - **How logical records fit into CIs,**
  - **How CIs fit onto tracks and CAs,**
  - **How much freespace is needed per CI (leaving enough freespace in each CI for one insert would use more freespace in smaller CIs)**
- **Generally, bigger DATA CI sizes are better than smaller ones.**

VSE/VSAM

VSE/VSAM Physical Data Storage...

KSDS Index Structure

- **Sparse Balanced Tree Index**
  - **Each index entry references high key of next lower level index or data CI**
  - **Only right-most index record at any upper index level is not filled**
  - **Only one high level index record exists, containing pointers to next lower index level or to data, if only one Control Area (CA) is in data set**
  - **An intermediate index level only exists if there are more lower level records than can be mapped in one higher level record**



VSE/VSAM

VSE/VSAM Physical Data Storage...

KSDS Index structure...

VSE/VSAM KSDS Low Level Index (Sequence Set) Control Interval (CI) SI-01

CI-01P CI-02P CI-03P CI-04P . . . FSCI FSCI . . . H-PTR

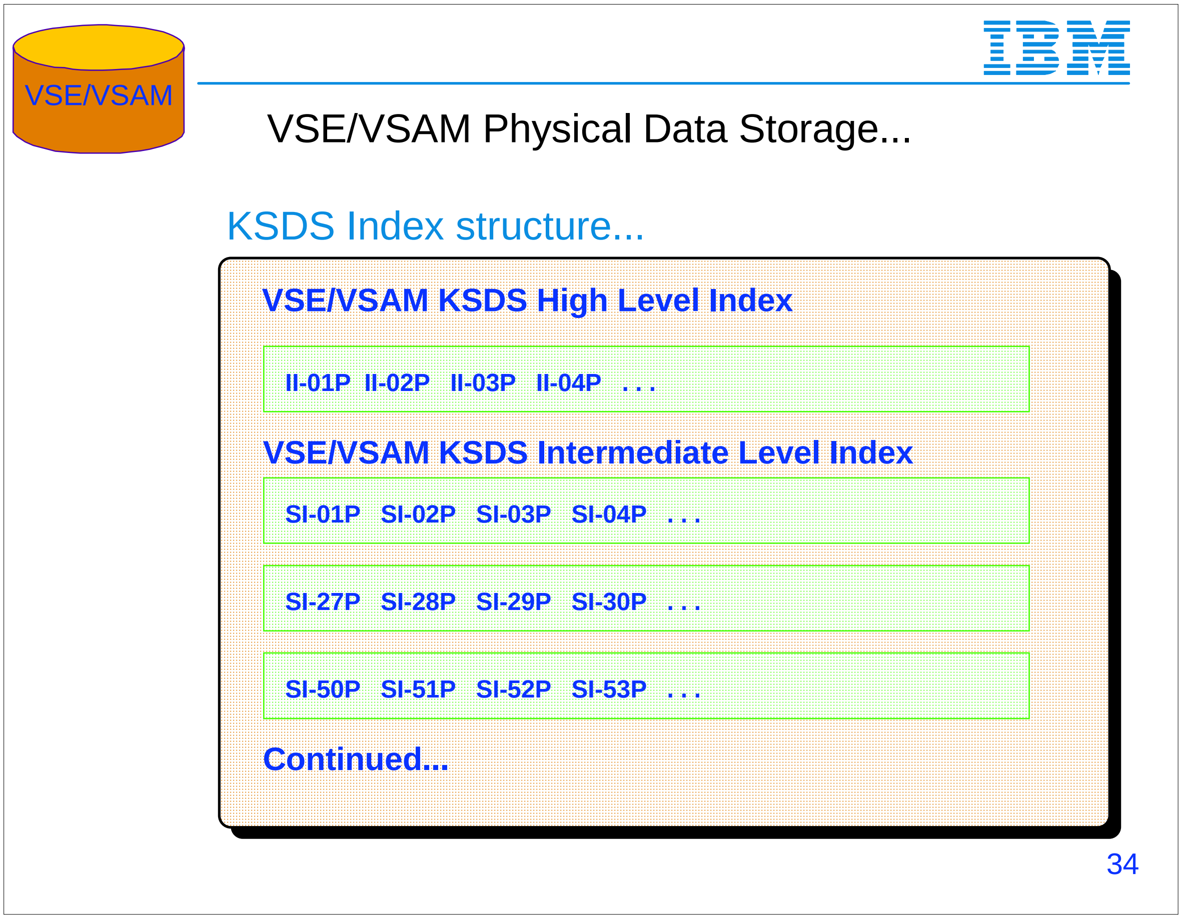
VSE/VSAM KSDS Data Control Area

CI-01 CI-02 CI-03 CI-04 CI-05 CI-06

CI-07 CI-08 CI-09 CI-10 CI-11 CI-12

CI-19 CI-20 CI-21 CI-22 CI-23 CI-nn

Continued...



VSE/VSAM

VSE/VSAM Physical Data Storage...

KSDS Index structure...

VSE/VSAM KSDS High Level Index

II-01P II-02P II-03P II-04P . . .

VSE/VSAM KSDS Intermediate Level Index

SI-01P SI-02P SI-03P SI-04P . . .

SI-27P SI-28P SI-29P SI-30P . . .

SI-50P SI-51P SI-52P SI-53P . . .

Continued...

VSE/VSAM Physical Data Storage...

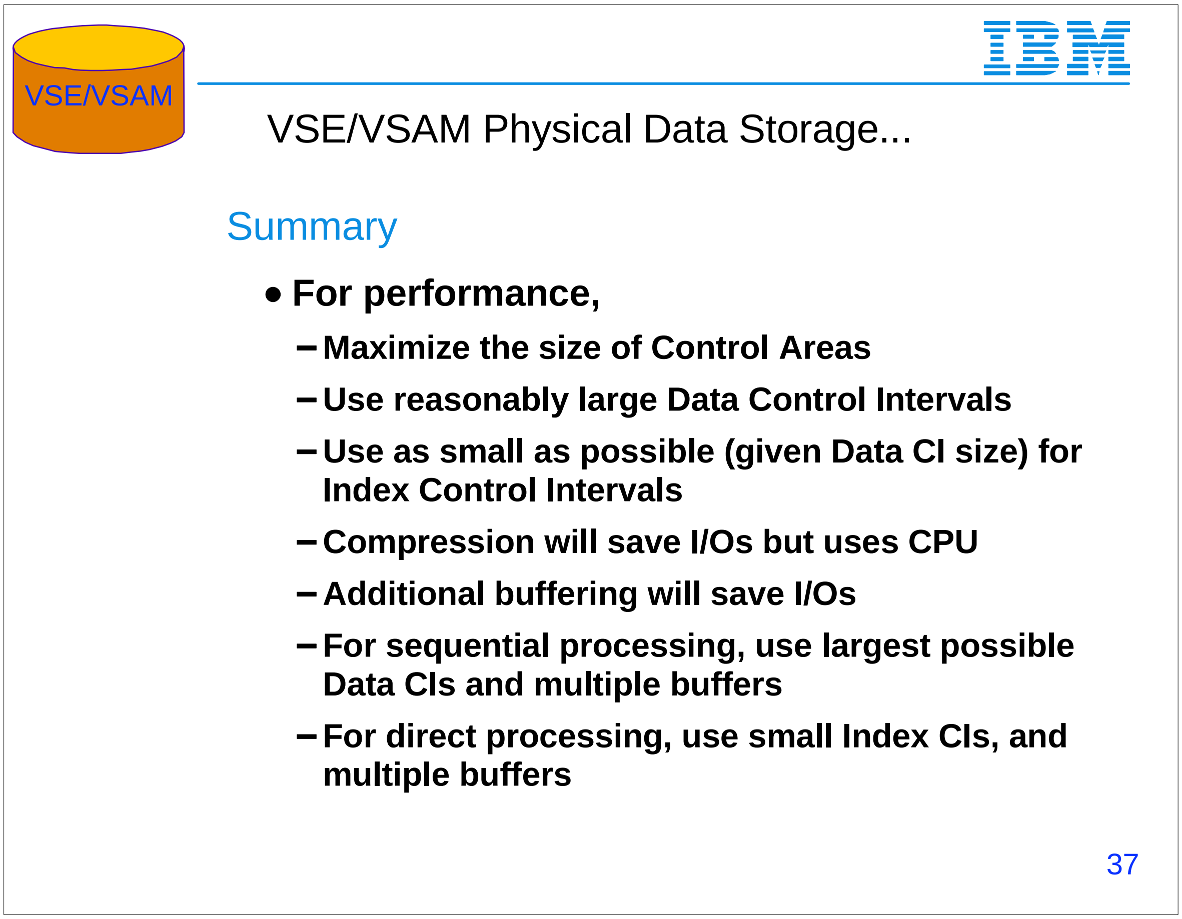
KSDS Index Structure...

- **Index entries at all levels are compressed**
- **First entry in an index record is uncompressed**
- **All entries contain:**
  - **Address of next level or data CI (2 bytes)**
  - **Compression information (3 bytes)**
  - **Full key or significant part of a compressed key**
- **Benefits**
  - **Saves disk, buffer space**
  - **Saves index I/O operations**

VSE/VSAM Physical Data Storage...

Examples for 3390 DASD:

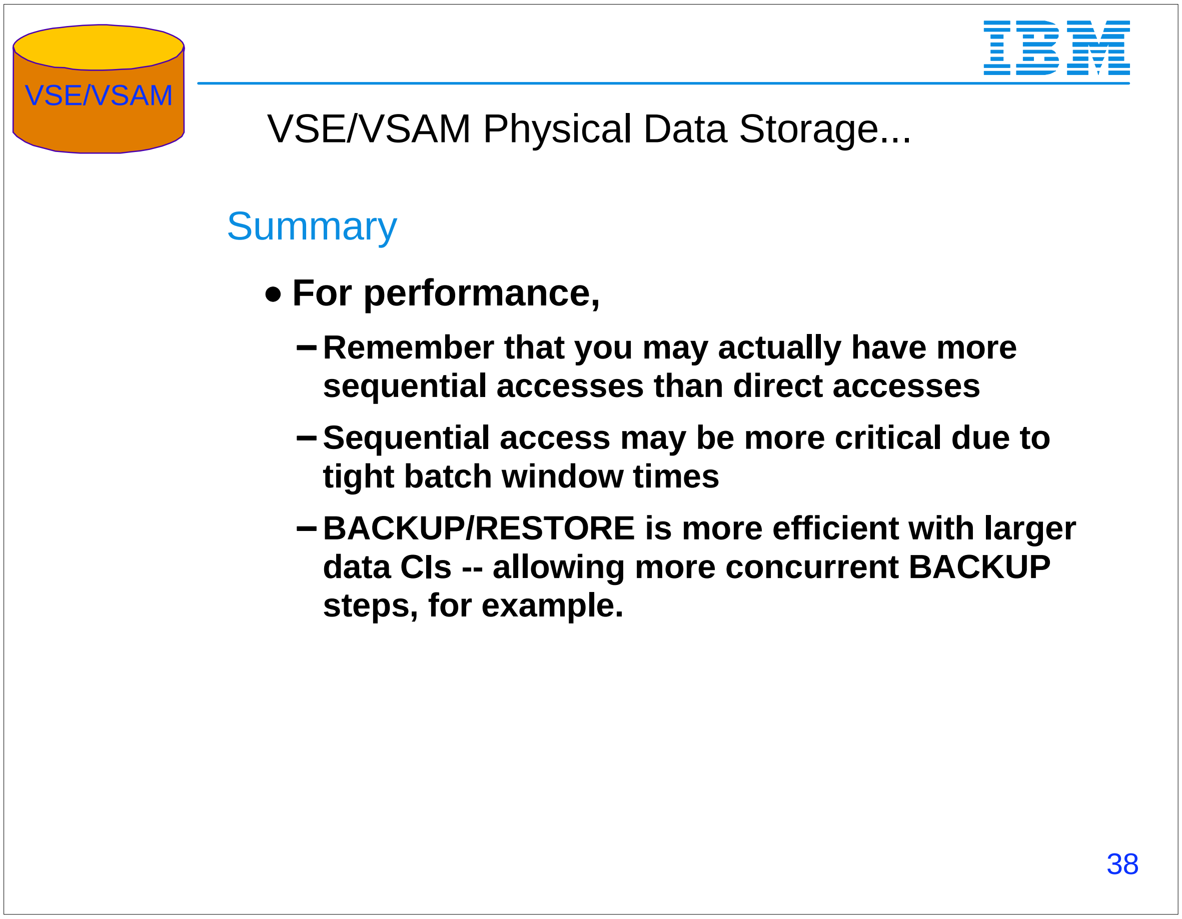
Data CI = 8K	Data CI = 2K
Index CI = 1K	Index CI = 4K
90 Data CI/CA	315 Data CI/CA
720K Data/CA	630K Data/CA
L2 ~ 100 CAs	L2 ~ 300 CAs
L3 ~ 10K CAs	L3 ~ 90K CAs
7.2 GB	56 GB
For 3 GB....	For 3 GB....
1 high level 1K	1 high level 4K
42 intermediate 43K	17 intermediate 72K
4,200 seq. set 4243k	4,821 seq. set 19356K
375K data CIs	1,518K data CIs



VSE/VSAM Physical Data Storage...

Summary

- **For performance,**
  - **Maximize the size of Control Areas**
  - **Use reasonably large Data Control Intervals**
  - **Use as small as possible (given Data CI size) for Index Control Intervals**
  - **Compression will save I/Os but uses CPU**
  - **Additional buffering will save I/Os**
  - **For sequential processing, use largest possible Data CIs and multiple buffers**
  - **For direct processing, use small Index CIs, and multiple buffers**



VSE/VSAM Physical Data Storage...

Summary

- **For performance,**
  - **Remember that you may actually have more sequential accesses than direct accesses**
  - **Sequential access may be more critical due to tight batch window times**
  - **BACKUP/RESTORE is more efficient with larger data CIs -- allowing more concurrent BACKUP steps, for example.**

VSE/VSAM Catalogs

Catalog Options and Requirements

- **One Master Catalog for a VSE/VSAM system**
  - **Required by system**
  - **Assigned during IPL (DEF CAT=) or when first DEFINE MASTERCATALOG is done (during installation)**
  - **In standard VSE/ESA system, contains:**
    - **Definition for Compression Control Data Set**
    - **Definition for VSE Messages Online file (required)**
    - **Definition for VSESP USER CATALOG and any user defined user catalogs**
    - **Definition for VSAM managed libraries - system and user defined**

VSE/VSAM Catalogs...

Catalog Options and Requirements...

- **User Catalogs**
  - **Optional -- as many can be defined as desired**
  - **JCL specified**
- **Catalog Residency Rules:**
  - **Catalogs can be shared by multiple VSE images**
  - **Only one catalog can reside on a logical volume**
  - **A catalog can own space on any number of volumes**
  - **Multiple catalogs can own space on one volume**

VSE/VSAM Catalogs...

Catalog Options and Requirements...

- **Catalog Contents**
  - **Self Describing Records**
  - **User Catalog Pointers**
  - **Volume and Space Definitions**
  - **Cluster Definitions**
  - **Data and Index Component Definitions**
  - **Alternate Index and Path Definitions**
  - **Recoverability Information**
  - **Compression Information (In Compression Control Data Set)**

VSE/VSAM Catalogs...

Catalog Options and Requirements...

- **Recommendations**
  - **Use application based naming**

    ```
    PAYROLL.TRANS.HIST
    PAYROLL.MASTER
    PAYROLL.TAX
    ```

  - **Name all cluster components explicitly**

    ```
    PAYROLL.MASTER (cluster)
        PAYROLL.MASTER.DATA (data component)
        PAYROLL.MASTER.INDEX (index comp.)
    ```

  - **Exploit partition and system independent naming**
  - **With RAMAC Virtual Array, consider more volumes and catalogs -- perhaps by application**

VSE/VSAM

VSE/VSAM Catalogs...

Catalog Options and Requirements...

- **<u>Recommendations...</u>**
  - **Use separate catalogs for**
    STATIC files
    (files seldom defined or deleted)
    DYNAMIC files
    (files frequently defined or deleted, or with frequent secondary allocations)
  - **Use separate catalogs for**
    Files needed by production on-line systems
    Files used only by batch
  - **<u>Don’t put all your eggs in one basket, or all your files in one catalog!</u>**

VSE/VSAM

VSE/VSAM Catalogs...

Catalog Recoverability

- **"Recoverable Catalogs"**
  - **Implemented using Catalog Recovery Areas (CRA)**
  - **Recoverable Catalog Exposures:**
    - If catalog is damaged, CRA can help recover
    - If CRA is damaged, catalog (and its files) can’t be opened
    - If CRA can be restored with data, catalog can be updated
  - **Recoverable Catalog Tools**
    - LISTCRA, RESETCAT, EXPORTRA, IMPORTRA
  - **Logic errors and recoverable catalogs**
    - Bad records are written to both catalog and CRA

VSE/VSAM

VSE/VSAM Catalogs...

Catalog Recoverability...

- **"Recoverable Catalogs"...**
  - **Recoverable catalogs were designed in the days of removable disks to help recover from head crashes or other media failures in the catalog**
  - **Recoverable catalogs slow catalog processing since all updates to the catalog are written to two separate disks -- this extends the window of opportunity for system failures**

Recommendation

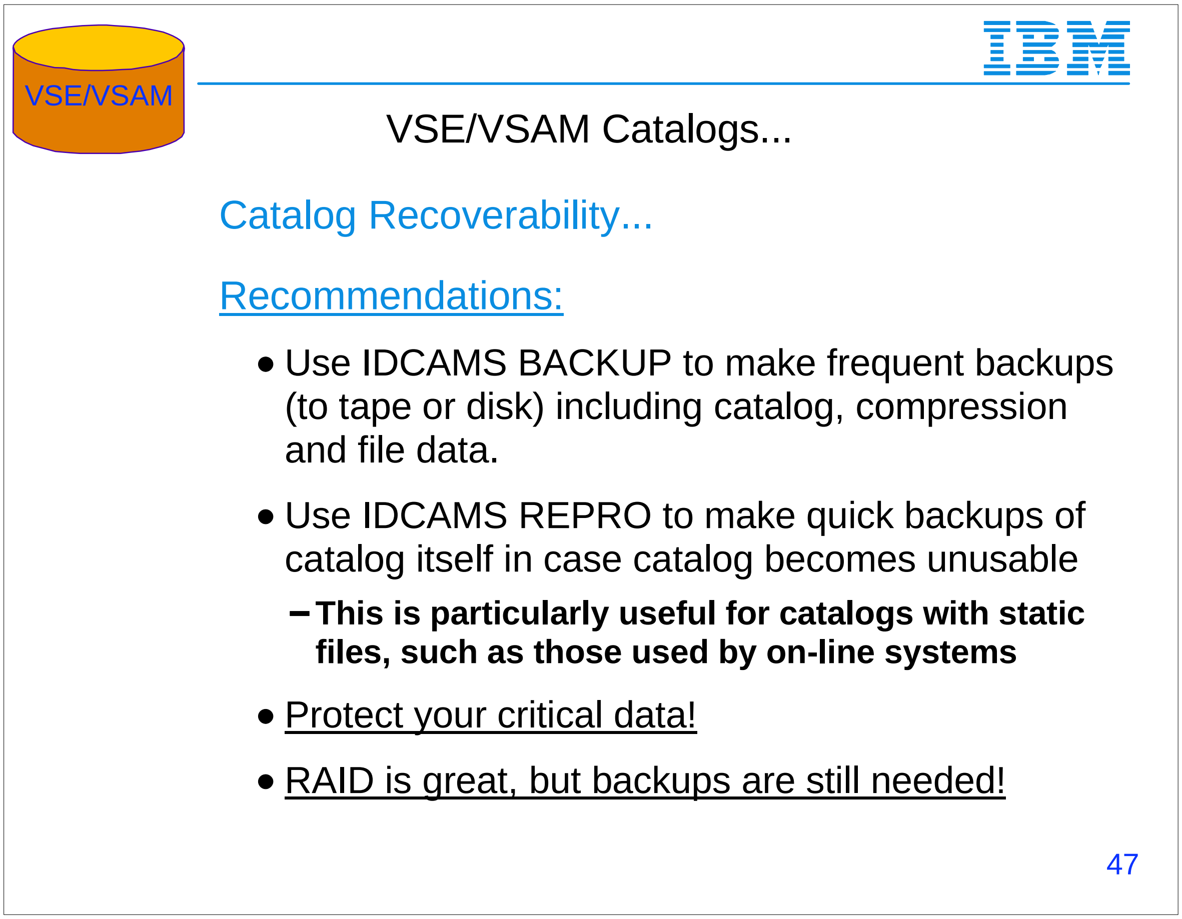
- **Do NOT use RECOVERABLE catalogs**

VSE/VSAM

VSE/VSAM Catalogs...

Catalog Recoverability...

- **"Non-recoverable Catalogs"**
  - **Recovery requires backup of data AND catalog definition (and compression) information**
  - **VSE/VSAM IDCAMS BACKUP/RESTORE copy data, compression, and catalog data**
  - **VSE/VSAM IDCAMS REPRO can backup just the catalog information (without the data)**
    - In many cases, this can recover current level data (instead of down level data) if only the catalog is damaged
    - Recovery can be quick in these cases
    - See VSE/VSAM Commands (REPRO command)



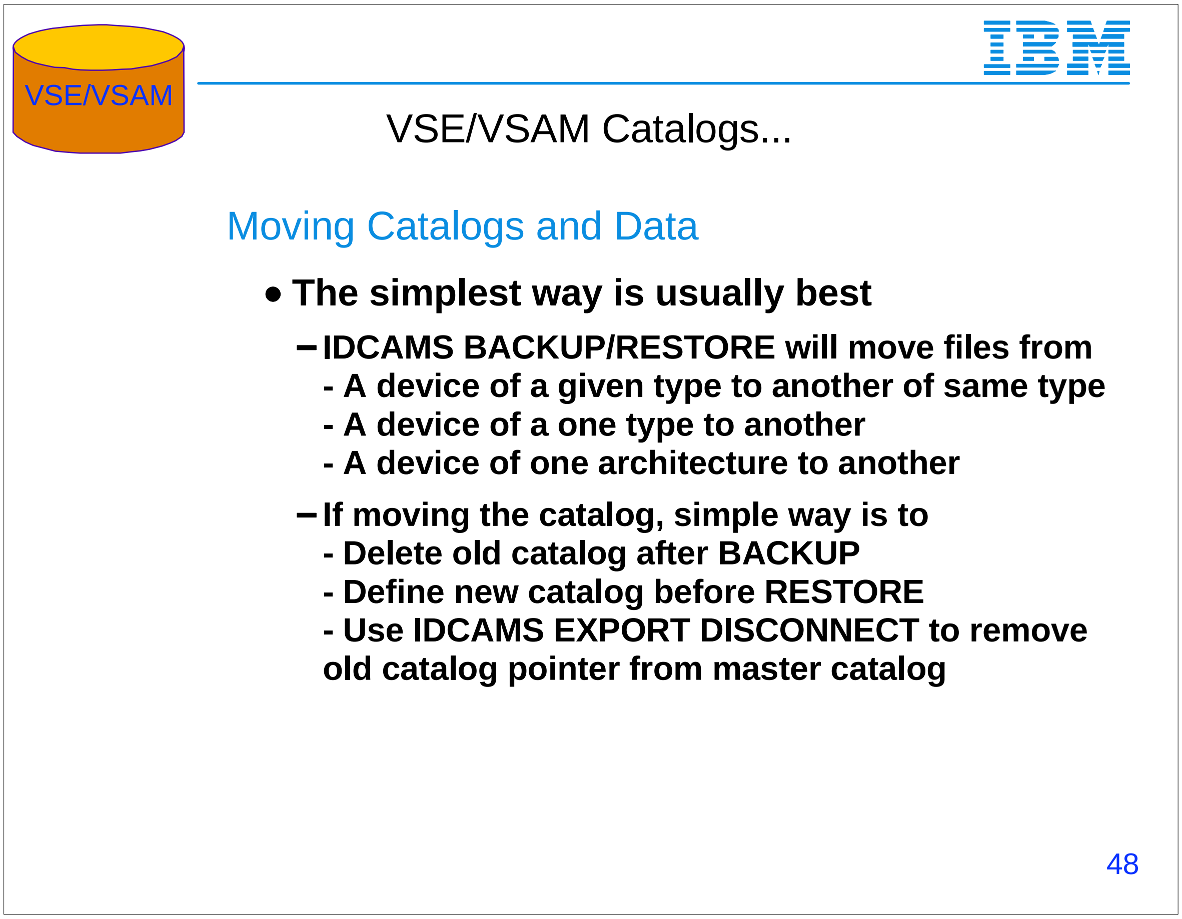
VSE/VSAM

VSE/VSAM Catalogs...

Catalog Recoverability...

Recommendations:

- Use IDCAMS BACKUP to make frequent backups (to tape or disk) including catalog, compression and file data.
- Use IDCAMS REPRO to make quick backups of catalog itself in case catalog becomes unusable
  - **This is particularly useful for catalogs with static files, such as those used by on-line systems**
- Protect your critical data!
- RAID is great, but backups are still needed!



VSE/VSAM

VSE/VSAM Catalogs...

Moving Catalogs and Data

- **The simplest way is usually best**
  - **IDCAMS BACKUP/RESTORE will move files from**
    - A device of a given type to another of same type
    - A device of a one type to another
    - A device of one architecture to another
  - **If moving the catalog, simple way is to**
    - Delete old catalog after BACKUP
    - Define new catalog before RESTORE
    - Use IDCAMS EXPORT DISCONNECT to remove old catalog pointer from master catalog

VSE/VSAM Catalogs...

Moving Catalogs and Data...

- **If changing device types or architectures often some changes are desired**
  - **Simplest approach:**
    BACKUP, DELETE, DEFINE, RESTORE (as above)
    - Then LISTCAT to examine results, especially for CI sizes if file used with CICS LSR pools
    - Reorganize logically for files where needed
  - **TRACKS/CYLINDERS vs BLOCKS changes are handled automatically with this approach**
  - **DEFINE CLUSTER statements need to be changed before reorganization**

VSE/VSAM Catalogs...

Moving Catalogs and Data...

- **If changing device sizes (e.g. 3390-1 to 3390-3)**
  - **Catalog definitions for the volume contain a bitmap whose size is set when the first space on the volume is defined.**
  - **FASTCOPY (etc.) to a larger volume does not change the bitmap.**
- **Recommendations**
  - **BACKUP all VSAM objects on the volume(s)**
  - **DELETE all VSAM objects on the volume(s)**
  - **DELETE CATALOG/SPACE on the volume(s)**
  - **DEFINE CATALOG/SPACE**
  - **RESTORE all VSAM objects on the volume(s)**

VSE/VSAM

VSE/VSAM DASD Space Allocation

Allocation takes time, virtual and real storage

- Allocation specification affects
  - Data Control Area (CA) size
  - Index Control Interval (CI) size
- Space allocation units
  - Records (General Purpose)
    - Record allocation uses average record size
  - Blocks (FBA, VFBA only)
  - Cylinders or Tracks (CKD, ECKD only)

VSE/VSAM

VSE/VSAM DASD Space Allocation...

Practical Factors

- Record allocation uses product of average record size and number of records
- Your space allocation amounts must include any defined free space
  - In an allocation of 5000 records with 20% CI and 20% CA free space, less than 3000 records can be loaded without another allocation
- Use LISTCAT after DEFINE to see whether what you got was what you wanted. Check again after loading

VSE/VSAM

IBM

VSE/VSAM DASD Space Allocation...

Recommendations

- Do NOT use small (less than cylinder or MAX-CA) allocations
- Check results of DEFINE and load using LISTCAT
  - **Data CI size (bigger = better)**
  - **Physical block size (= CI size ideally)**
  - **Index CI size (small as possible, rounded up to the next LSR buffer size for files used by CICS)**
  - **Number of CIs per CA**
  - **Size of CA (One CA for small files, cylinder size for larger files)**

VSE/VSAM

IBM

VSE/VSAM DASD Space Allocation...

Migrating to devices with larger track capacity

- **IDCAMS may increase index CI size to use CA space fully (if estimated to be too small)**
- **Migration:**
  - **From 3380 to 3390**
  - **Data CI Size 4K (does not change)**
  - **Index CI Size 2K (changes to 2.5K)**
  - **2.5K index CIs use same LSR pool buffers as 4K data -- causing poorer performance until retuned**
  - **Consider reorganizing the file with a larger data CI size (e.g. 8K) allowing smaller index CIs. LSR pool must be retuned**

VSE/VSAM Insertion -- CI and CA Splits

VSAM's Strategy for Insertions

- **Keep records logically in key sequence**
- **Keep records reasonably blocked**
- **Minimize need for index maintenance**
- **Retain sequential and direct performance**
- **Use available free space in neighborhood**
- **Create additional free space if needed**

This leads to CI and CA Split processing

VSE/VSAM Insertion -- CI and CA Splits

Control Interval (CI) Splits

- **A CI split occurs when insufficient free space exists in the assigned CI to accomodate an insert into that CI**
- **Processing**
  - **Set Split-In-Process bit in CIDF, write CI**
  - **Move higher key half of records to new buffer**
  - **Write new CI**
  - **Write updated sequence set with new pointer**
  - **Remove duplicated records from old CI in buffer**
  - **Write old CI with S-i-P flag off**

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

Control Interval (CI) Splits...

- **System fault**
  - **There will be temporarily duplicated data on disk**
  - **CI Split data integrity...**
    - **If S-i-P was found on during subsequent processing, complete the split**
    - **Duplicated data is erased when the partially split CI is re-processed**
- **No space (free CI) available**
  - **A CA Split becomes necessary**

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

Control Interval (CI) Splits...

- **Performance Implications**
  - **Watch LISTCAT CI SPLIT statistics**
  - **Tune by adjusting**
    - **CI Freespace**
    - **Reorganization frequency**
  - **CI Splits cost**
    - **A little CPU time during the split**
    - **A few (4) I/Os during the split**
    - **A little extra DASD space**
    - **A little extra processing after the split**

VSE/VSAM Insertion -- CI and CA Splits...

Control Interval (CI) Splits...

- **Recommendations**
  - CI splits are relatively inexpensive when they occur and during subsequent processing
  - If inserts are infrequent and clustered, CI splits may perform better than distributed free space
  - If inserts are frequent and uniform (over the time span of reorganization) some distributed free space may be better than CI splits
  - DO NOT reorganize because some magic number of CI splits has happened -- check the rate at which they are still happening

VSE/VSAM Insertion -- CI and CA Splits...

Control Area (CA) Splits

- **A split occurs when there is no free CI in a CA and a CI split is required to create free space for an insertion**
- **Processing**
  - Set Split-In-Process bit in Sequence Set, write CI
  - Format new CA at the High Used RBA
  - Move the higher-keyed half of CIs to new CA (read and write)
  - Write new Sequence Set CI
  - Update higher level index records
  - Write updated sequence set with new pointer
  - Remove duplicated CIs from old CA (write)
  - Write old Sequence Set CI with S-i-P flag off

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

Control Area (CA) Splits...

- **System fault**
  - **Temporarily duplicated data**
  - **CA Split data integrity...**
    - If S-i-P was found on during subsequent processing, complete the split
    - Duplicated data is erased when the partially split CA is re-processed
- **No space for new CA**
  - **Take new secondary allocation**
- **No space available for new allocation**
  - **Reject insert request with NOSPACE**

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

Control Area (CA) Splits...

- **Performance Implications**
  - **Watch LISTCAT CA Split statistics**
  - **Tune by adjusting**
    - CA Free Space
    - Reorganization Interval
  - **CA Splits cost**
    - Lots of CPU time during the split
    - Lots (many hundreds) of I/Os during the split
    - Extra DASD space (two CAs are half full)
    - Some extra processing after the split (the effective blocking factor has been decreased)

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

Control Area (CA) Splits...

- **Recommendations**
  - **CA splits are quite expensive when they occur but do not significanly impact subsequent processing**
  - **If inserts areheavily clustered, CA splits may perform better than distributed free space**
  - **If inserts are frequent and uniform (over the time span of reorganization) distributed free space may be better than CA splits**
  - **Consider preformatting files rather than splitting when the file must be "loaded" in direct mode**
  - **DO NOT reorganize just because a magic number of CA splits have occurred.**

VSE/VSAM

VSE/VSAM Insertion -- CI and CA Splits...

CI and CA Split "Dos and Don’ts"

- **Don’t worry about CI splits**
- **Don’t reorg just because some splits occur**
- **Do make backups (just don’t restore them)**
- **Don’t reorg if tomorrow’s inserts may use the free space created by today’s splits**
- **Define CA free space to handle some CI splits**
- **Watch CI/CA split statistics**
  - **Before deciding on reorganization interval**
  - **Does number of new splits decline after a few days?**
- **Reorg squeezes out free space splits have inserted**
  - **future splits may have to put it back**

VSE/VSAM Insertion -- CI and CA Splits...

General comments about splits

- Insertions are the problem, not splits
- Frequent reorganization has several bad aspects
  - It takes time to reorganize files
  - It removes free space created by splits
  - It probably causes the split rate (splits/day?) to increase, if inadequate free space was defined
- Reorganizing at a certain split count is usually not a good idea.

Performance Factors

Control Interval (CI) Size tips

- New DASD Architectures (non-synchronous, RAID, RVA...)
- Cached DASD
  - Bigger Data CIs means smaller Index CIs
  - Bigger Data CIs may mean fewer CI splits with a given free space percentage
  - NOIMBED, NOREPLICATE
- LSR Considerations
  - Set Data CI to 4K, 8K, 12K...(8K good start)
  - Set Index CI to smallest possible LSR pool size

Performance Factors...

Control Areal (CA) Size tips

- **CA Size should be as big as possible -- it is the largest of:**
  - **Cylinder or MAX-CA size**
  - **Primary allocation amount**
  - **Secondary allocation amount**
- **Recommendations**
  - **Make primary and secondary amounts at least a cylinder (MAX-CA)**
  - **Use NOIMBED and NOREPLICATE (defaults)**

Performance Factors...

Control Areal (CA) Size tips...

- **Recommendations...**
  - **How to, with RECORDS allocation**
    - Estimate number of records in CI
    - Estimate number of CIs in MAX-CA
    - Multiply (# of records) * (CI/CA)
    - Set primary and secondary allocation amounts some multiple of the product
  - **Simpler way to do this**
    - DEFINE CLUSTER with trial value
    - LISTCAT, review CA Size
    - Alter DEFINE CLUSTER if needed to have suitable CI/CA size values

Performance Factors...

CI and CA Size tips...

- **Remember**
  - **Most VSAM defaults designed when 2314 was king**
  - **Not optimum for modern equipment**
- **Recommendations:**
  - **Use bigger DATA CI sizes**
  - **Trading virtual and real storage saves CPU cycles and I/O operations**
  - **Today, NOIMBED and NOREPLICATE are right**
- Cache effects
- ECKD improvements
  - **No IMBED benefits with cache, or adequate BUFNI**

Performance Factors...

File Load (and Reorganization) Time Tips

- **SPEED vs. RECOVERY**
  - **RECOVERY is the default**
    - CAs are pre-formatted before records are loaded
    - Permits restart of user written load program
  - **SPEED is the preferred option**
    - Only affects initial load after DEFINE
    - Bypasses performat during initial load
    - No effect after inital load
    - Saves 1/3 to 1/2 of I/O operations during load

Performance Factors...

File Load (and Reorganization) Time Tips...

- **FREESPACE**
  - **Only applicable during sequential insert**
    Initial load, sequential add of new records
  - **Excessive free space ("bubbles" in the data)**
    - **Extra I/O operations**
    - **Less efficient buffering and caching**
    - **Slower normal (sequential and direct) processing**
  - **Too little free space**
    - **Extra split activity**
    - **Slower insert processing**
    - **Slower normal (sequential and direct) processing**

Performance Factors...

File Load (and Reorganization) Time Tips...

- **FREESPACE Pathological Cases**
  - **Clustered inserts**
    - **Preload Strategy**
      Preload extra records in active key ranges
      Close file, reopen, delete extra records
    - **Alter FREESPACE Strategy**
      Code subroutine to call IDCAMS during load
      Load records with normal free space
      Close, call subroutine to alter free space
      Reopen, load additional records
      Repeat alter free space and load as needed
      to define free space appropriately

Performance Factors...

File Load (and Reorganization) Time Tips...

- **FREESPACE Pathological Cases...**
  - **Dummy record load, all inserts done directly**
    - **- Predictable Keyrange Strategy**
      - **Load about right number of dummy records in right key ranges**
      - **Close, reopen, and then delete all records**
    - **- Constant Keyrange Stragegy**
      - **Do not DELETE/DEFINE cluster**
      - **Just delete records**
  - **Reorganize <u>LESS</u> often**
    - **- Let splits create free space where inserts occur**

7Performance Factors...

STRINGs and Shared Resources

- **STRINGs are concurrently active place holder**
  (~1 K storage, in VSE/ESA V2, 31-bit eligible)
- **Normal (Non-Shared Resources) NSR**
  - **Typical batch processing**
    - **- Buffer lookaside only in buffers for string**
      - **Desired record could be in another string's buffers**
    - **- Sequential read-ahead and write-behind**
- **Online (Local Shared Resources) LSR**
  - **Default CICS File handling**
    - **- Buffer lookaside across all buffers holding file CIs**
    - **- No sequential read-ahead nor write-behind**
    - **- Pool of placeholders and buffers for all files in LSR pool**
    - **- Saves I/O operations, related CPU time costs**

Performance Factors...

Buffers, Buffers, and MORE Buffers

- **Index processing - Direct**
  - **Per string, or available in LSR Pool**

    Unacceptable...

    one buffer (old default)

    Acceptable...

    one buffer per index level or 3,
    whichever is greater (new default)

    Good...

    One buffer for each high level and mid-level index record, plus one per string

    Best...

    One buffer for each index record

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Index processing - Sequential**
  - **Per string, or available in LSR Pool**

    Good and Best...

    One buffer (default)
    Two buffers (if insert activity)

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Data processing - Direct**
  - **Per string, or available in LSR Pool**

    Acceptable

    One buffer (default)

    Two buffers if insert activity

    Better

    More buffers (watch LSR statistics, EXCP statistics) depending on locality of reference and probability of reuse of CI contents
  - **Successful lookasides > 10 times buffer reads**

    (see next page)

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Data processing - Direct (New LSR considerations)**
- **VSE/ESA 2.4 and earlier:**

  Successful lookasides > 10 times buffer reads

  and limit number of buffers in subpool to 150 (or less) -- if more buffers needed, split files to new subpools. More buffers OK if single file uses the subpool.
- **VSE/ESA 2.5 and later:**

  Successful lookasides >> 10 times buffer reads

  There is no longer a need to restrict buffers within a subpool, as the LSR Buffer Hashing algorithm has removed the sequential buffer search
- **See next pages**

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **VSE/VSAM LSR Buffer Hashing**
  - **Buffer search overhead reduced by eliminating sequential search through the buffer pool. New algorithm is a hashing technique. Processor cycles formerly used to search buffers are now free for application use, and allow more efficient use of larger capacity processors.**
  - **Processor cycle path length is now consistent regardless of the number of buffers. That is, search time is now independent of the buffer pool size.**

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **VSE/VSAM LSR Buffer Hashing -- Implementation**
  - **Hash table, used as anchor for BCB search**
  - **Synonym handling through BCB chaining**
  - **Hashing Algorithm:**

```
A = Rem(RBA/2 + AMBaddr/2)/(2N - 1)

A   -- offset into hash table
Rem -- remainder function
RBA -- Relative Byte Address (or CI num)
AMB -- VSAM control block unique per DS
N   -- number of buffers in pool
```

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Example**

Hash Table

0	1	2	3	4	5	6	7	8	9	10

Buffer Control Blocks

Location	RBA	Synonym	Buffer Pointer
100			
200			
300			
400			
500			
600			

6 BCBs = 6 buffers in pool. N = 6,
(2N - 1) = 11 = size of hash table
Assume AMB address = 0 for this example

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Example...**

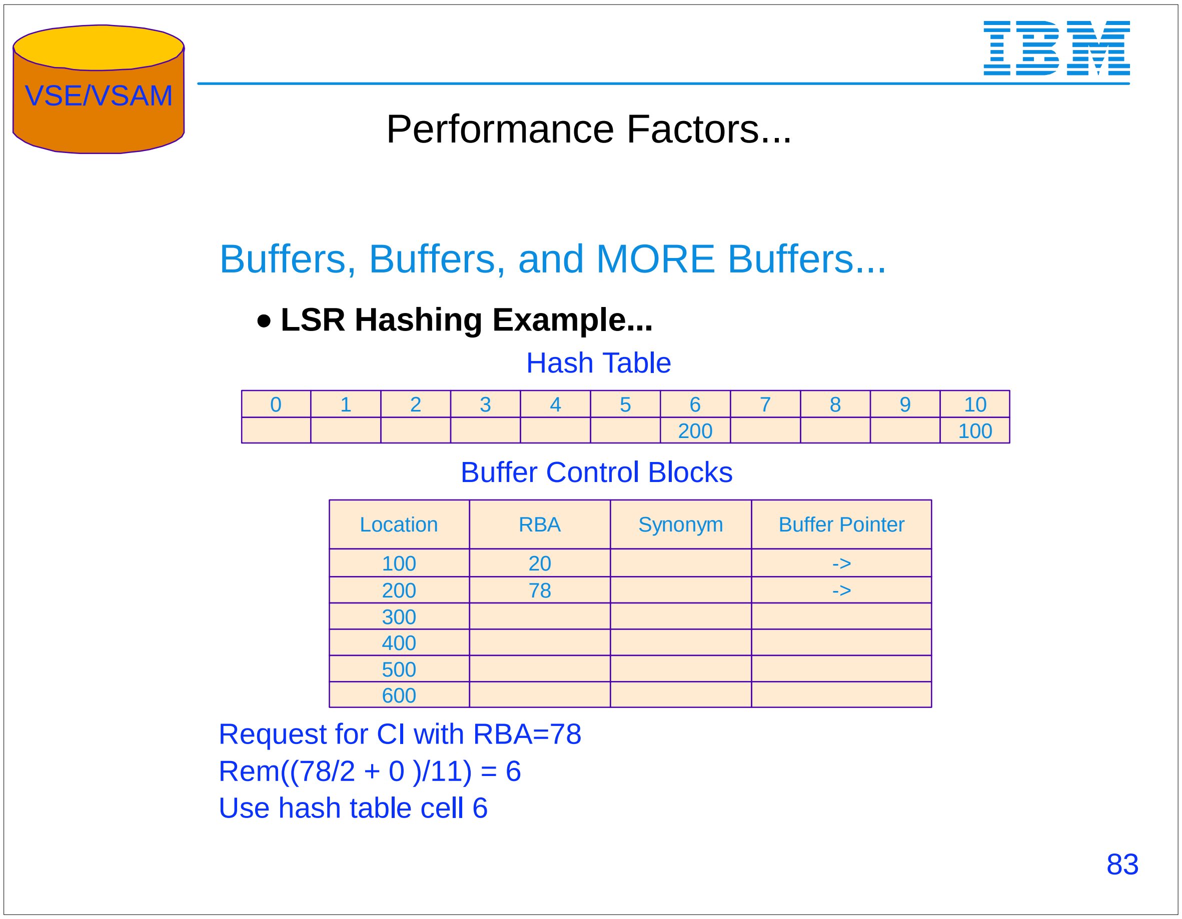
Hash Table

0	1	2	3	4	5	6	7	8	9	10
										100

Buffer Control Blocks

Location	RBA	Synonym	Buffer Pointer.
100	20		->
200			
300			
400			
500			
600			

Request for CI with RBA=20
Rem((20/2 + 0)/11) = 10
Use hash table cell 10.



VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Example...**

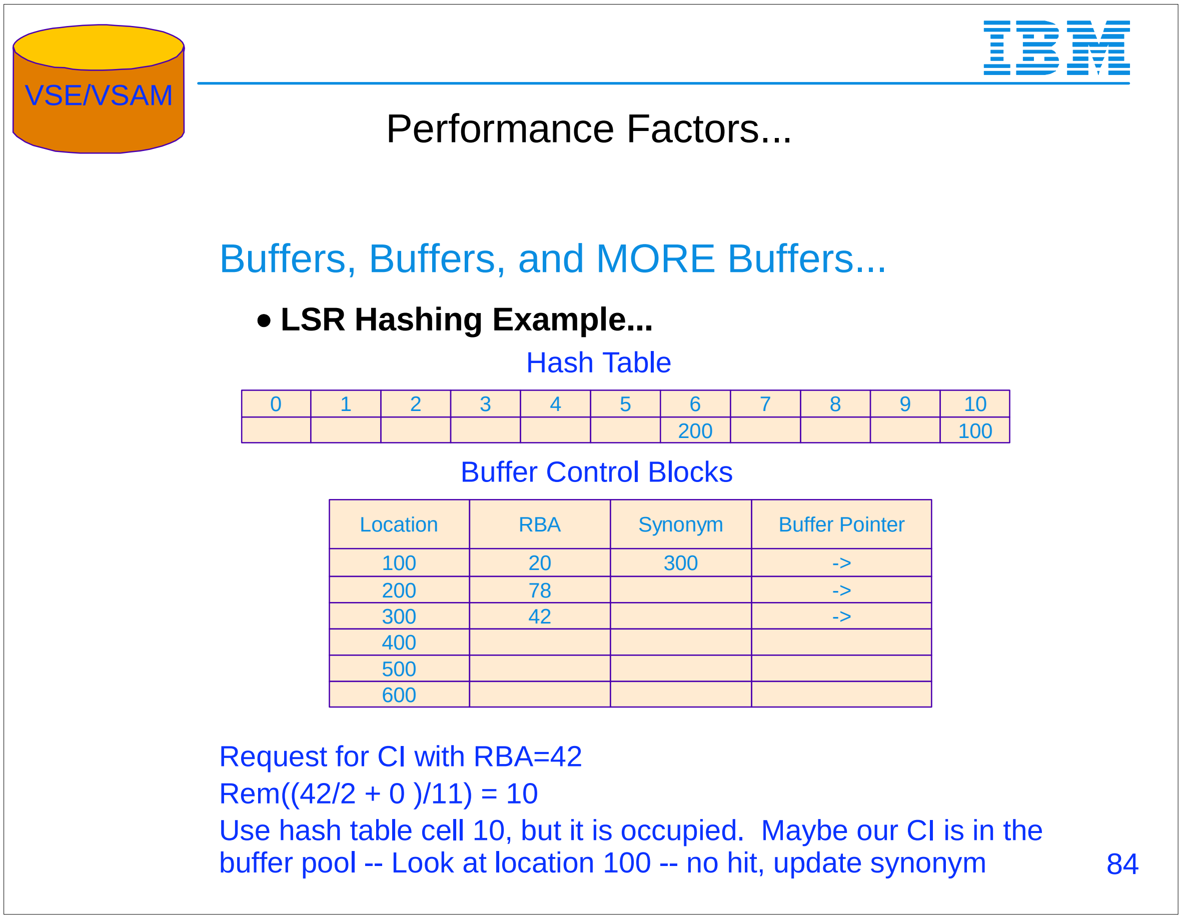
Hash Table

0	1	2	3	4	5	6	7	8	9	10
						200				100

Buffer Control Blocks

Location	RBA	Synonym	Buffer Pointer
100	20		->
200	78		->
300			
400			
500			
600			

Request for CI with RBA=78
Rem((78/2 + 0)/11) = 6
Use hash table cell 6



VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Example...**

Hash Table

0	1	2	3	4	5	6	7	8	9	10
						200				100

Buffer Control Blocks

Location	RBA	Synonym	Buffer Pointer
100	20	300	->
200	78		->
300	42		->
400			
500			
600			

Request for CI with RBA=42

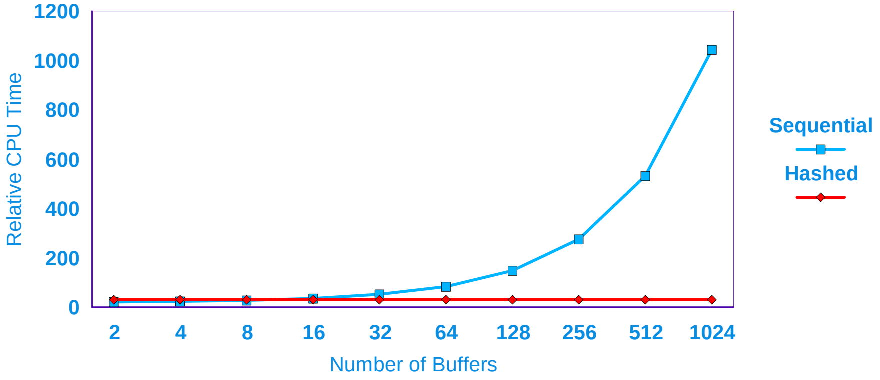
Rem((42/2 + 0)/11) = 10

Use hash table cell 10, but it is occupied. Maybe our CI is in the buffer pool -- Look at location 100 -- no hit, update synonym

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Theory**



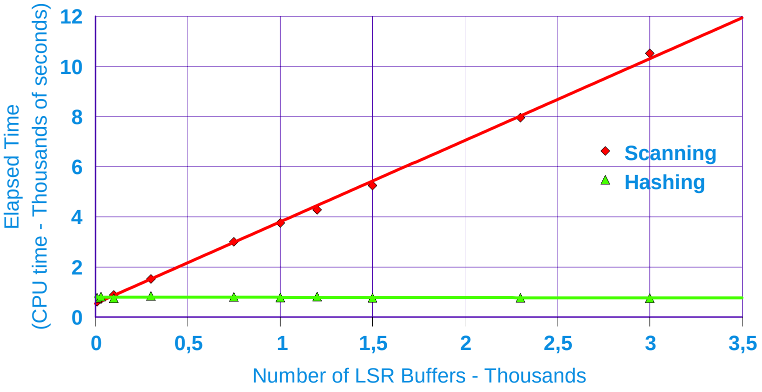
Note X axis is logarithmic

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Results**

VSAM Random Read

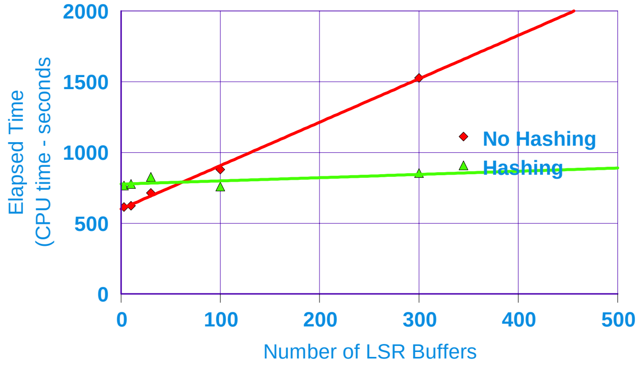


Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **LSR Hashing Results**

VSAM Random Read



Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Data processing - Sequential NSR**
  - **Chained I/O strategy used**
    optimized for sequential batch processing
  - **Chain-read-ahead or -write-behind**
  - **No overlap of CPU and I/O**
  - **With large virtual and real storage availability**
    - specify large number of data buffers
    (BUFND=nn)

3380/9345, 4 K CIs	**1 cylinder = 150 CIs**
3390, 8K CIs	**1 cylinder = 90 CIs**

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Data processing - Sequential LSR**
  - **No read-ahead, no chained I/O**
    - Far from optimum for sequential batch
  - **Consider Sequential Processing Benefits**
    - Whenever input and/or output can be sorted
    - Rather than an AIX Bowse
    This is a sequential read of AIX pointers
    Then a direct read of base cluster records
    - Even if only a few logical records are processed per physical read.

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies**
  - **LISTCAT EXCP Statistics**
    - Do a LISTCAT for data set before and after test
    - Statistics are cumulative, look at the difference
  - **$JOBACCT exit statistics**
    - ICCF Library 59 sample routine prints data
  - **CICS Shutdown Statistics**
    - Recent CICS (2.2 and later) show LSR statistics
  - **VSE/ESA Examples sample subroutine**
    - Usable for batch, etc.

VSE/VSAM

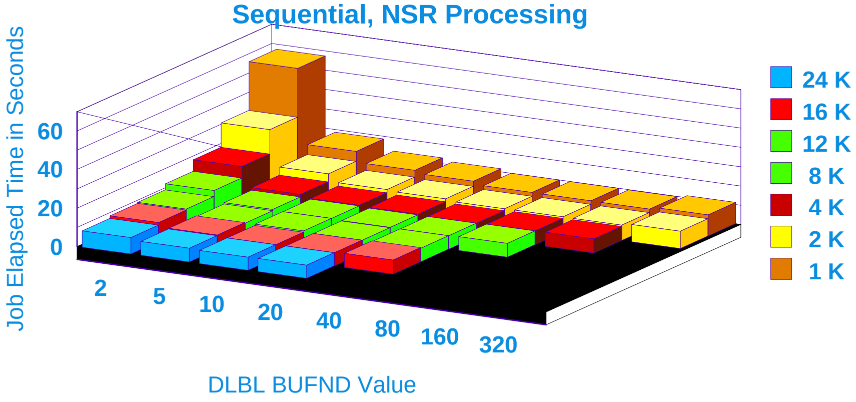
IBM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies**

File Load Time
vs. CI Size and BUFND
Sequential, NSR Processing



VSE/VSAM

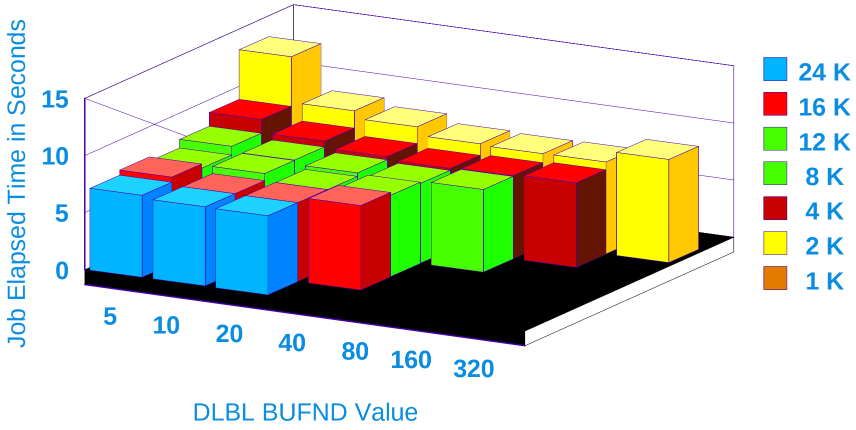
IBM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies...**

File Load Time
vs. CI Size and BUFND
Sequential, NSR Processing



VSE/VSAM

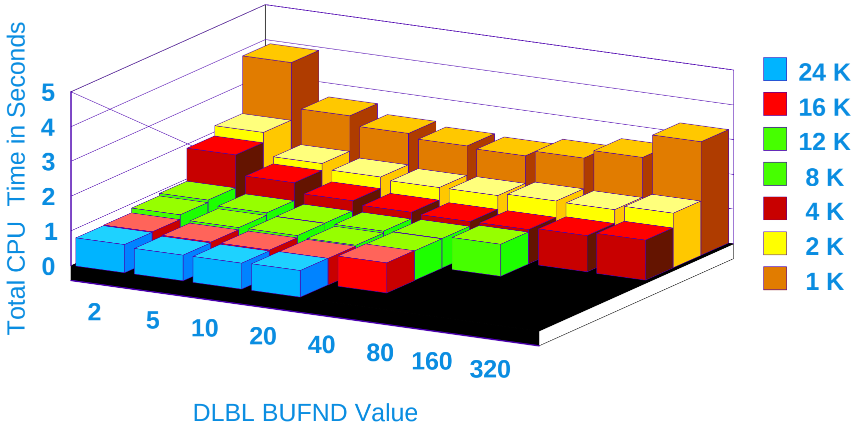
IBM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies...**

Total CPU Time
vs. CI Size and BUFND
Sequential, NSR Processing



VSE/VSAM

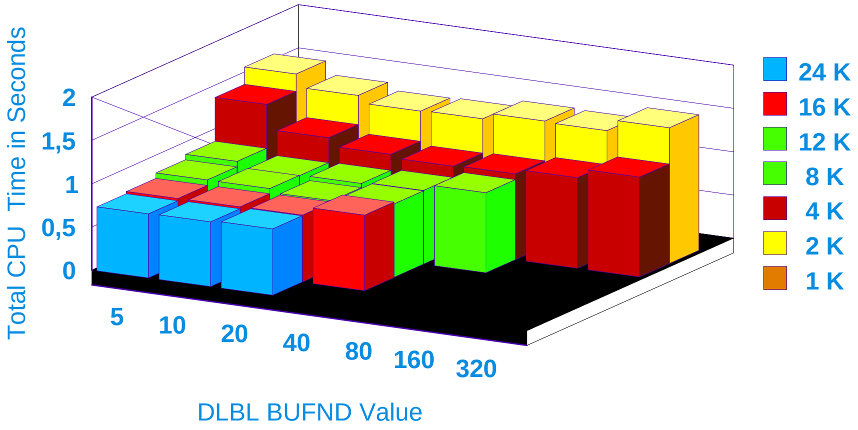
IBM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies...**

Total CPU Time
vs. CI Size and BUFND
Sequential, NSR Processing



VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies...**

The above cases were created using IDCAMS DEFINE CLUSTER and DITTO BV. Measurement used SKJOBACC (ICCF library 59) to capture the elapsed time and CPU Time data.

Test Parameters:
9345 file load, 4000 records each 1000 bytes in length, initial load,
SPEED option set in DEFINE CLUSTER.

VSE/VSAM

Performance Factors...

Buffers, Buffers, and MORE Buffers...

- **Evaluation of Buffering and I/O Strategies...**
  - **Think big, but do not start paging**
  - **Saving process synchronous file I/O while increasing paging I/Os is a bad idea**
  - **Other factors to consider**
    - CPU time spent searching long LSR pools
    BETTER to have more pools with fewer buffers per pool than fewer pools with more buffers per pool to get the same hit ratio
    - CPU time spent managing I/O operations may be bigger than you think
    Especially if a guest of VM!

VSE/VSAM

VSE/VSAM Data Set Sharing
(The Good, The Bad, and The Ugly)

First Principles of sharing

- **"VSAM Sharing is not a performance option"**
  - **Dan Janda, 1972**
- **"TANSTAAFL"**
  - **Robert A. Heinlein, *The Moon Is A Harsh Mistress,* 1966**
- **VSE Sharing based on Lock Manager and File**
  - **VM, OS/390 implementations do not use lock file**
- **No system provided integrity for sharing with VM and OS/390 systems**

VSE/VSAM

VSE/VSAM Data Set Sharing...

First Principles of sharing...

- **Access means opening, processing, closing**
- **VSE and VSAM permit concurrent access**
- **VSE and VSAM protect against concurrent update damage**
- **<u>Correct user specifications are required</u>**
  - **SHAREOPTIONS(nn)**
  - **DASD Volume IDs**
  - **VSE ADD statements**
  - **Lock File definition**

VSE/VSAM

VSE/VSAM Data Set Sharing...

First Principles of sharing -- Levels of Integrity:

- No Integrity
  - VSE and VSAM do NOT protect your data. Your data can be destroyed.
- Write Integrity
  - VSE and VSAM will protect your data against being corrupted by other output activity.
- Read Integrity
  - VSE and VSAM will ensure your application always sees the latest version of any record, regardless of other activities in your system.

VSE/VSAM

VSE/VSAM Data Set Sharing...

First Principles of sharing -- Scope of Protection:

- VSE/VSAM uses VSE Lock Manager services
- Within system sharing
  - VSE Lock Table (in storage) is used
    - File on disk device NOT added with ADD ...,SHR
- Across system sharing
  - VSE Lock File on disk is used
    - File on disk device added with ADD ...,SHR
    - Real disk if multiple real CPUs, or LPARs
    - VM Virtual disk if all VSE systems are in one VM

VSE/VSAM Data Set Sharing...

First Principles of sharing -- Scope of Protection...

- **CAUTION**
  - **All shared DASD MUST be defined with ADD ...,SHR**
  - **All shared DASD MUST have unique VOLIDs for all volumes known to all systems using one VSE Lock File**
  - **These are NOT system enforced**
  - **Violation leads to "unpredictable" but BAD results**
    - **Lock hangs**
    - **Lost data!**

VSE/VSAM Data Set Sharing...

Shareoption (SHR) levels

- **SHAREOPTION(1)**
  - **External locks at OPEN (data set level)**
- **SHAREOPTION(2)**
  - **External locks at OPEN (data set level)**
- **SHAREOPTION(3)**
  - **No locks**
- **SHAREOPTION(4)**
  - **External locks at OPEN (data set level)**
  - **Internal locks at request (CI/CA level)**
- **SHAREOPTION(4 4)**
  - **External locks at OPEN (data set level)**
  - **External locks at request (CI/CA level)**
- **If data set not on shared volume, external lock becomes internal**

VSE/VSAM Data Set Sharing...

SHAREOPTION(1)

- **Scope -- intra and inter system sharing**
- **Overhead -- OPEN time only locking**
- **Functions**
  - **Write Integrity**
  - **Read Integrity**
- **Concurrency**
  - **Any one output OPEN -- or --**
  - **Any number of input OPENs**

VSE/VSAM Data Set Sharing...

SHAREOPTION(2)

- **Scope -- intra and inter system sharing**
- **Overhead -- OPEN time only locking**
- **Functions**
  - **Write Integrity only**
- **Concurrency**
  - **One output OPEN -- and --**
  - **Any number of input OPENs (at the same time)**

VSE/VSAM Data Set Sharing...

SHAREOPTION(3)

- **Scope -- intra and inter system sharing**
- **Overhead -- (NO Locking)**
- **Functions**
  - **NO Integrity -- read or write**
  - **NO VSAM provided Integrity services**

    Programs which open a data set for output concurrently will cause loss of data, including loss of data previously inserted by other programs, even if only one of the programs actually updates data.
- **Concurrency**
  - **Any number of OPENs without regard to intent**

VSE/VSAM Data Set Sharing...

SHAREOPTION(4)

- **Scope --**
  - **Intra-system -- multiple OPENs for output**
  - **Inter-system -- other systems OPEN for input only**
- **Overhead --**
  - **OPEN time only external locking**
  - **Request time locking of CI and CA accesses**

    CI locked at READ for update time
    CA locked at WRITE time
    unlock and end-of-request processing
  - **CI contents invalidated at end of request**

    Forced refresh of buffer contents
    No read-ahead in sequential processing

VSE/VSAM Data Set Sharing...

SHAREOPTION(4)...

- **Functions**
  - **Write Integrity**
  - **Read Integrity for output OPEN users**
    A read-only program may have pre-fetched records into its buffers which were deleted or updated by an update program after they were read, but before they were processed by the read-only program
- **Concurrency**
  - **Any number of output OPENs from one system**
  - **Any number of input OPENs from other systems**

VSE/VSAM Data Set Sharing...

SHAREOPTION(4 4)

- **Scope --**
  - **Intra-system -- multiple OPENs for output**
  - **Inter-system -- multiple OPENs for output**
- **Overhead --**
  - **OPEN time external locking**
  - **Request time external locking of CI and CA accesses (as above for SHR(4))**
  - **CI contents invalidated at end of request**
    - forces refresh of buffer contents
    - No read-ahead in sequential processing

VSE/VSAM Data Set Sharing...

SHAREOPTION(4 4)...

- **Functions**
  - **Write Integrity**
  - **Read Integrity for output OPEN users**
    A read-only program may have pre-fetched records into its buffers which were deleted or updated by an update program after they were read, but before they were processed by the read-only program
- **Concurrency**
  - **Any number of output OPENs from any number of sharing systems**

VSE/VSAM Data Set Sharing...

Performance Implications

- **Significant performance degradation going from:**
  - **SHR(1) or SHR(2) going to SHR(4)**
  - **SHR(4) going to SHR(4 4)**
- **Degradation comes from:**
  - **Additional CPU (processor) time**
  - **Additional I/O operations**

VSE/VSAM Data Set Sharing...

Recovery Concerns

- **CICS Recovery/Restart processing**
  - **CICS cannot see changes by other jobs**
  - **Emergency Restart may delete other changes**
  - **Use CICS Transaction Server EXCI if needed**
- **Read-only jobs may see old data**
  - **Including VSE/VSAM Control Information**
  - **ABENDs may occur, or incorrect output**
- **DL/I Recovery/Restart considerations**
  - **No functional support in VSAM or DL/I for sharing via VSAM Data Set Sharing**
  - **Use DL/I MPS support if needed**

VSE/VSAM Alternate Indexes

AIX -- a special KSDS

- **Record key - alternate key from base cluster**
- **Record data - pointer to base cluster record**
- **Non-unique alternate keys are supported**
  - **Records may be long due to multiple pointers**
  - **Spanned record can be up to a CA in size**
    sets limit to number of records with a common key
- **AIXes in the upgrade set are automatically updated when the base cluster is updated**
- **No provision for sparse indexing**

VSE/VSAM Alternate Indexes...

PATH -- an AIX/Base Cluster structure

- **PATH entry through AIX permits**
  - **alternate key access**
  - **a NOUPDATE option**
  - **access through path is transparent**
- **PATH buffers and control blocks**
  - **Additional virtual and real storage**
  - **Read integrity (for multi-path access)**
  - **PATH buffers apply to AIX**
  - **Default (catalog) buffers apply to base cluster**

VSE/VSAM Alternate Indexes...

Performance Considerations

- **Access through a path is**
  - **Direct or sequential access to the AIX records**
  - **Direct access to the base cluster records**
- **Buffering for base cluster access**
  requires catalog BUFFERSPACE coding
- **Consider NOUPDATE and rebuild of AIX**
  instead of dynamic update
- **Consider sequential extract and sort**
  of base cluster records (without using AIX) if many records are touched (e.g. 1/10 or more); AIX access if few are touched (1/1000 or less). In between -- try it and see what works best.

VSE/VSAM Alternate Indexes...

Consider building your own AIX

- **Maybe not all records need AIX pointer**
- **NOUPGRADE option may be useful**

Alternate Index Record Layout

A	**B**	**C.....**	**D**	**E...........**	**F...**	**F....**	**F....**	**F.... (continued...)**

A	Pointer type x’00’ = RBA, X’01’ = Primary Key
B	Pointer length
C	Number of pointers in record (halfword)
D	Alternate key length
E	Alternate key from base cluster (key of this record)
F...	Pointers ("C" pointers, each "B" bytes long)

VSE/VSAM Alternate Indexes...

Consider building your own AIX...

- **These records are normally**
  - **built by IDCAMS BLDINDEX**
  - **maintained automatically by VSE/VSAM**
- **Can be**
  - **built by vendor or user program**
  - **maintained by VSE/VSAM or by user program**
- **Multiple pointers exist**
  - **if multiple base records contain the same alternate key (NONUNIQUEKEY)**
- **Problem when too many base records with same key**
  - **Largest record size = (DataCISize - 10) * (# CIs/CA)**

VSE/VSAM

VSE/VSAM Alternate Indexes...

CICS and Alternate Indexes

- **CICS Multi-path Access**
  - **CICS sees multiple files**
  - **CICS enqueues based on file name and key**
  - **Applications may see different levels of data**
  - **CICS cannot protect data**
    Recovery/Restart issues -- AIX may need recreation
  - **Circumventions:**
    - SHR(1) won’t work
    - SHR(2) only good solution
    - SHR(3) -- don’t call me!
    - SHR(4) very slow, may not fully protect
  - **Code all updates through one path**
    - All other paths defined as read-only

VSE/VSAM

VSE/VSAM Alternate Indexes...

CICS and Alternate Indexes...

- **CICS Newer features**
  - **CICS 1.7**
    - AIX Integrity (backout at recovery or restart
  - **CICS/VSE 2.1**
    - Can use LSR pools for AIXes
  - **CICS/VSE 2.2 (VSE/ESA 1.3)**
    - DATA SET NAME SHARING
    - All files sharing same data sets are linked in FCT
    - CICS provides read integrity to all views
    - Application programs (and users) see GOOD data
    - Check fix for APAR DY43264 or followon is applied

Recent Enhancements to VSE/VSAM

Many users have not yet exploited

- **VSE/VSAM 2.2 (VSE/ESA 1.3)**
  - **Buffers in GETVIS above 16MB line**
  - **Up to 16 LSR pools (CICS -- 15)**
    - **Frequently used files**
      can be assigned to different subpools
      compete less for I/O buffers
    - **Heavy browse files**
      can be separated from other files
  - **Significant relief of CICS DSA constraints**
  - **Reduced CPU time per transaction**
    - **Program storage compression reduced**
    - **Fewer physical I/Os needed**
  - **CICS Shutdown Statistics include VSAM LSR data**

Recent Enhancements to VSE/VSAM...

Many users have not yet exploited...

- **VSE/VSAM 6.1 (VSE/ESA 2.1/2.2 Cent. Funct. 6.1)**
  - **Additional control blocks move above the 16MB line**
    - **Place Holders (saves about 1KB per string)**
    - **Buffer Control Blocks (about 100 bytes per buffer)**
  - **All new and some old code moved to 31-bit SVA**
    - **Compression code is loaded in SVA-31**
  - **Many IDCAMS phases now SVA(-24) eligible**
    - **Not generally recommended**
    - **Probably better uses of SVA-24 storage**
  - **Check for PTFs UD49914, UD50055**
    - **SECTVAL Caching, Compression**
  - **Overhead with ICCF (DY44172 for Turbo or later)**
  - **OPEN Overhead -- need IKQVEX01 user exit in SVA (see APAR DY44066)**

VSE/VSAM and Cached DASD

Important Performance Tips

- **REPLICATE and IMBED should NOT be used**
  - **Index records are repeated around a track as often as they will fit**
  - **They're all the same, and all get cached!**
  - **That's a waste of cache storage**
  - **Even without cached DASD, specify enough**
    - BUFNI for batch
    - Index-sized buffers in LSR pool
    to avoid Index I/O operations as much as possible
- **Benefit -- fewer real I/Os, less CPU time**
- **Regular Data Format processing requires NOIMBED**

VSE/VSAM and Virtual Disks

VSE/VSAM can use Virtual Disks

- **Data in Memory for <u>unchanged</u> applications**
  - **Compiler workfiles**
  - **Job-to-job data-passing files**
- **Caution -- exploiting D-i-M speeds jobs compared to real disks**
  - **Less elapsed time**
  - **Higher utilization percentages**
  - **Lower priority jobs will run slower**

VSE/VSAM and Virtual Disks...

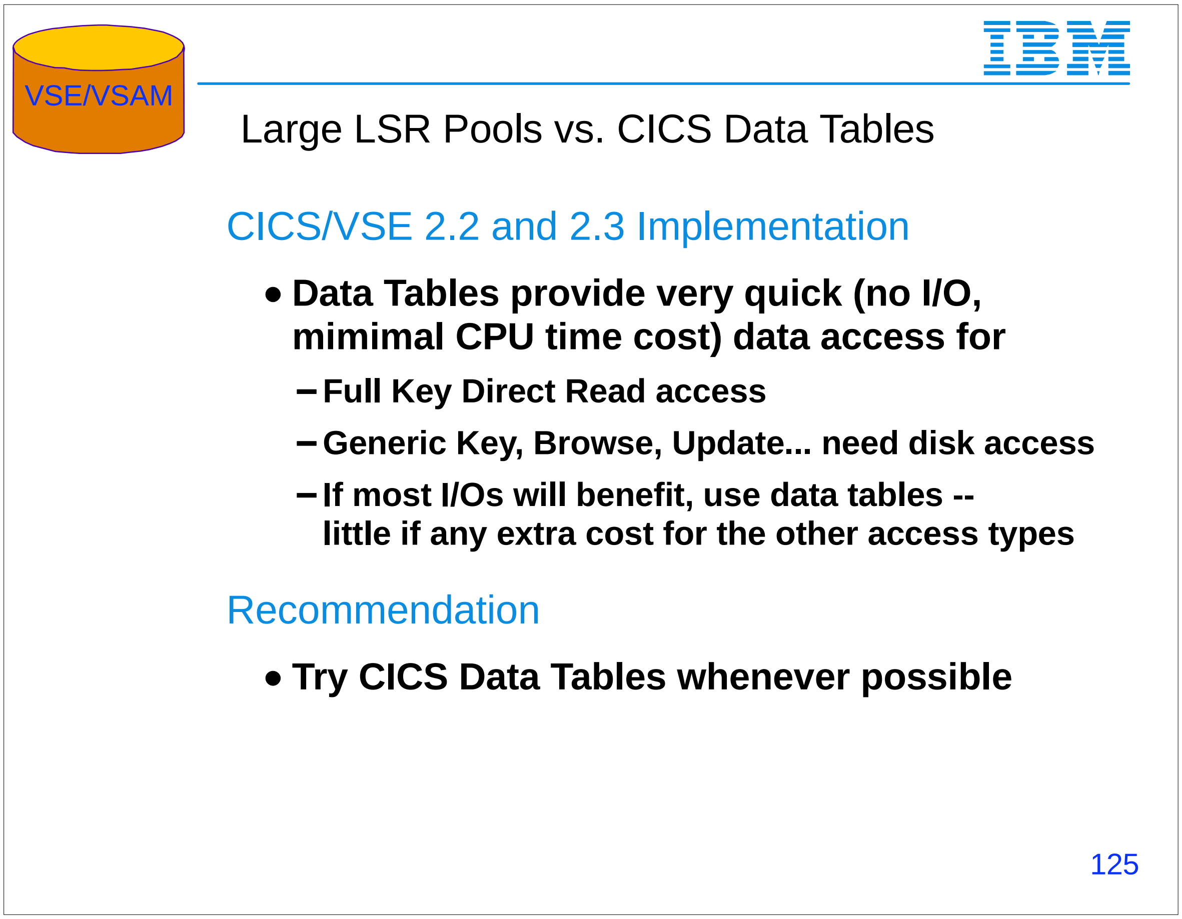
Compile Scenario

- **Background**
  - **User ran 3 compiles using VDisks, balanced**
  - **Payroll job in lower priority partition**
  - **When 4th compile started, payroll stopped!**
- **Recommendation**
  - **Recognize jobs using VDisks are more CPU intensive**
  - **Alter relative priorities of jobs**

VSE/VSAM and Virtual Disks...

Sort Scenario

- **Background**
  - **Sort workfiles are great candidate for VDisk**
  - **Most sorts are small, only a few are very large**
  - **Large partition virtual storage --**
    most sorts do not spill to disk
- **Recommendation**
  - **Moderate VDisk allocation for SORTWK1, regular disk allocations for SORTWK2..n**
  - **For IBM DFSORT, use**
    Large Partition GETVIS for smallest sorts,
    Dataspace for mid-sized sorts,
    Disk allocations for all SORTWKn files.



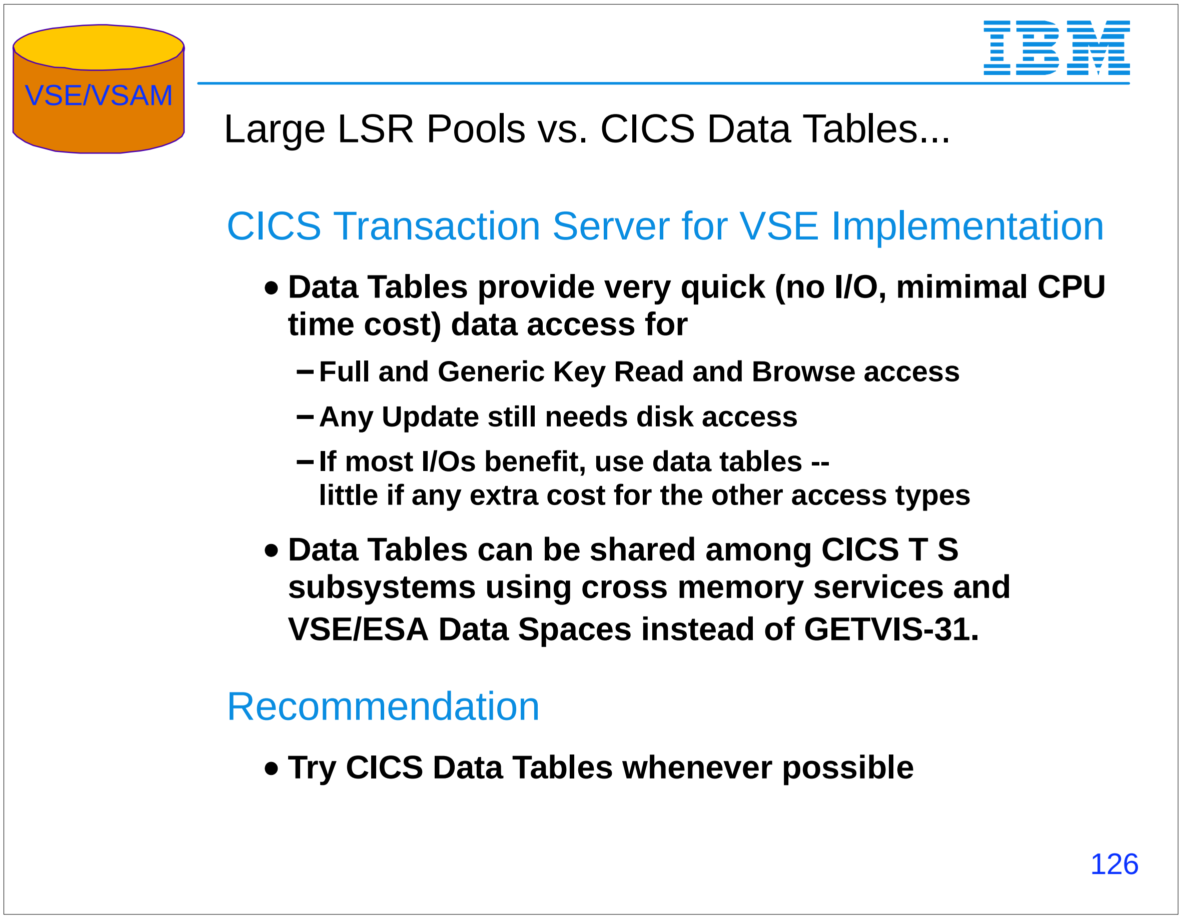
Large LSR Pools vs. CICS Data Tables

CICS/VSE 2.2 and 2.3 Implementation

- **Data Tables provide very quick (no I/O, mimimal CPU time cost) data access for**
  - **Full Key Direct Read access**
  - **Generic Key, Browse, Update... need disk access**
  - **If most I/Os will benefit, use data tables -- little if any extra cost for the other access types**

Recommendation

- **Try CICS Data Tables whenever possible**



Large LSR Pools vs. CICS Data Tables...

CICS Transaction Server for VSE Implementation

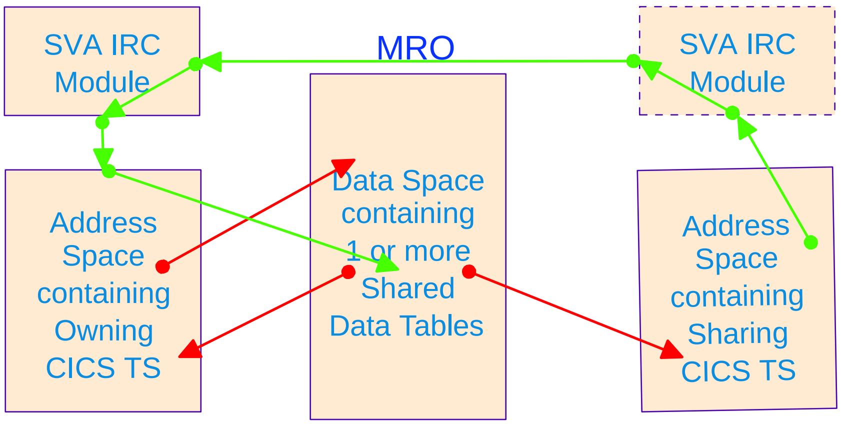
- **Data Tables provide very quick (no I/O, mimimal CPU time cost) data access for**
  - **Full and Generic Key Read and Browse access**
  - **Any Update still needs disk access**
  - **If most I/Os benefit, use data tables -- little if any extra cost for the other access types**
- **Data Tables can be shared among CICS T S subsystems using cross memory services and VSE/ESA Data Spaces instead of GETVIS-31.**

Recommendation

- **Try CICS Data Tables whenever possible**

Large LSR Pools vs. CICS Data Tables...

CICS Transaction Server for VSE Implementation



SVA IRC Module

MRO

SVA IRC Module

Address Space containing Owning CICS TS

Data Space containing 1 or more Shared Data Tables

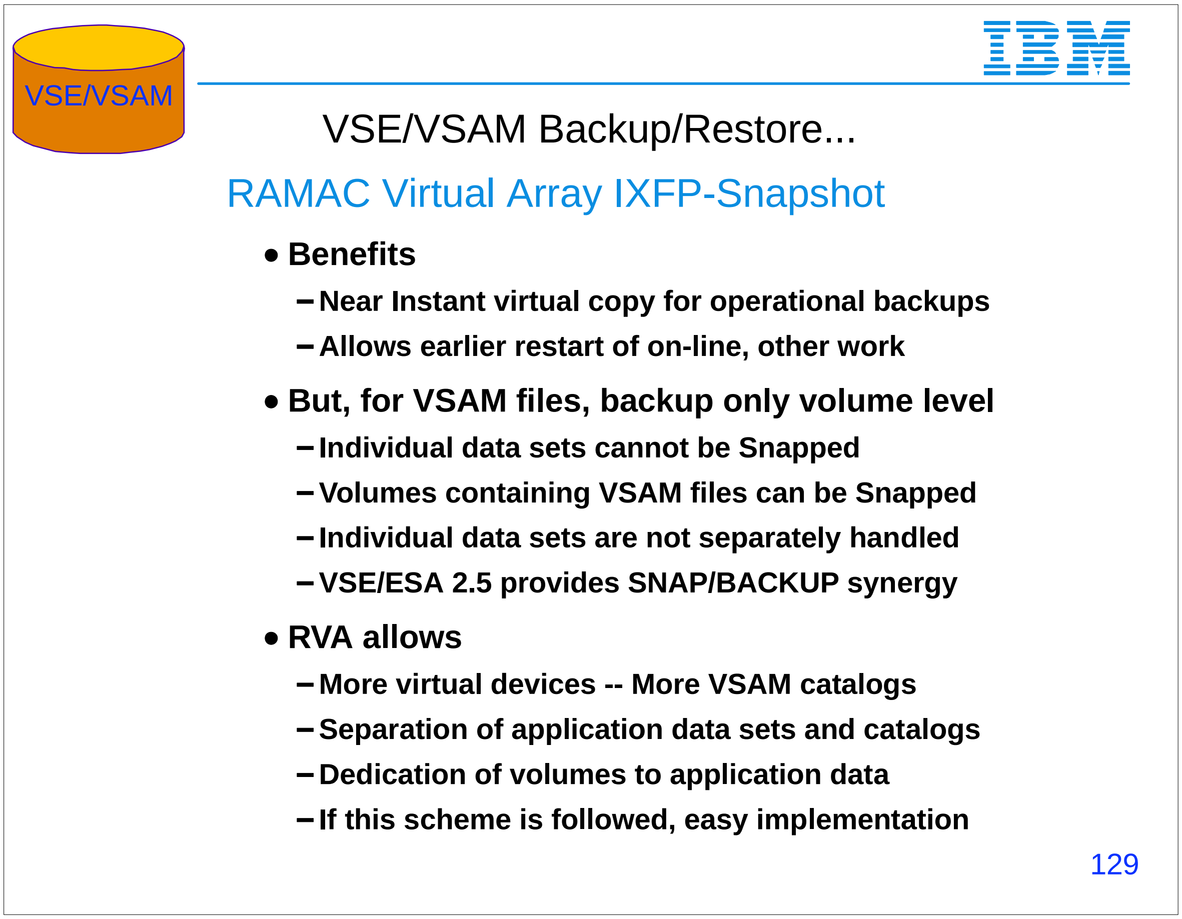
Address Space containing Sharing CICS TS

Note Read-for-update follows same path as shown above for update writes.

VSE/VSAM Data Compression

Compression - New in VSE/ESA 2.1

- **Uses S/390 Hardware Compression if available Compatible software routine if not**
  - **Effectiveness -- customer experiences show**
    - minimum compression of 20%
    - maximum compression of 80%
    - typically 40% to 60%
  - **Performance -- Hardware compression**
    - About one "equivalent instruction" per byte expanded
    - About three "equivalent instructions" per byte compressed
    - Plus setup for request
  - **Performance -- Software compressions**
    - About five times more CPU needed than H/W
  - **PTF UD49914 or later for good performance**

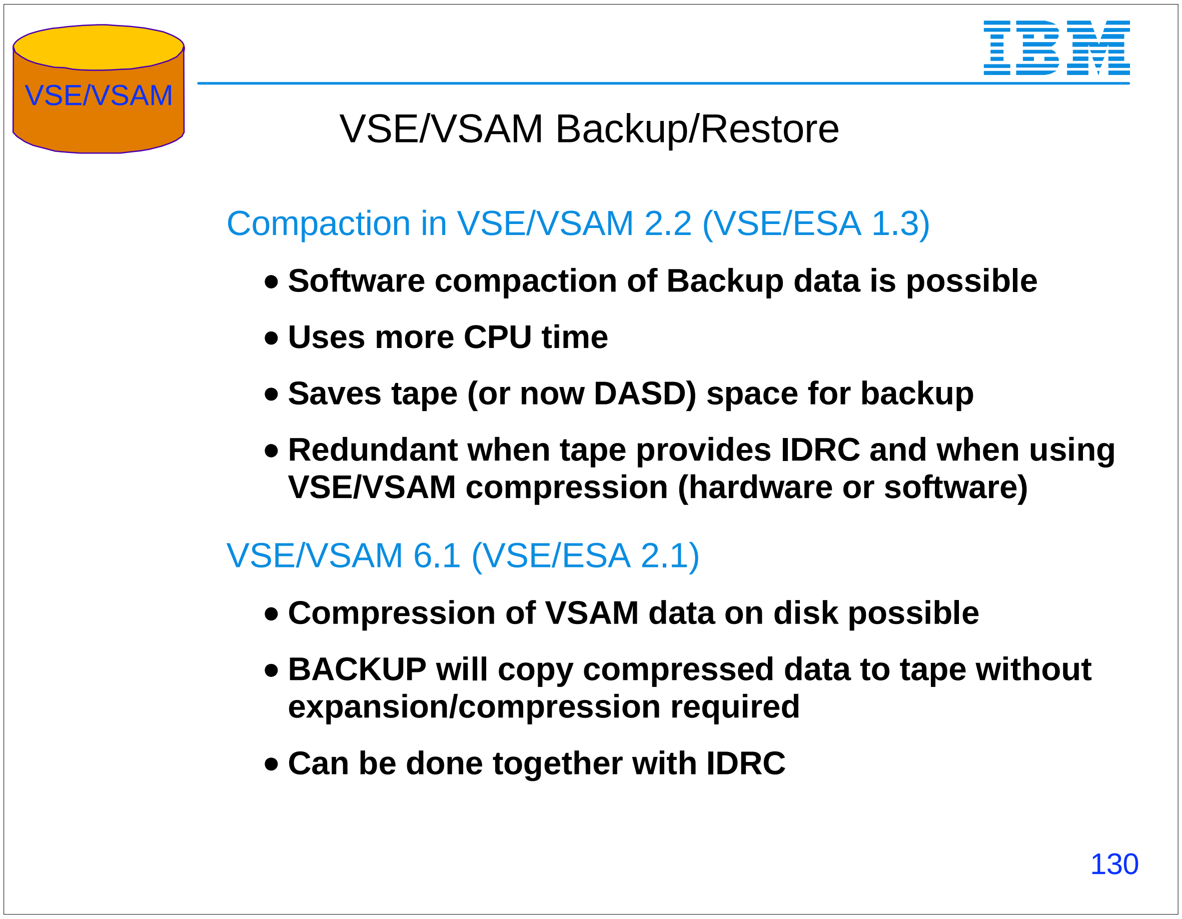


VSE/VSAM

VSE/VSAM Backup/Restore...

RAMAC Virtual Array IXFP-Snapshot

- **Benefits**
  - **Near Instant virtual copy for operational backups**
  - **Allows earlier restart of on-line, other work**
- **But, for VSAM files, backup only volume level**
  - **Individual data sets cannot be Snapped**
  - **Volumes containing VSAM files can be Snapped**
  - **Individual data sets are not separately handled**
  - **VSE/ESA 2.5 provides SNAP/BACKUP synergy**
- **RVA allows**
  - **More virtual devices -- More VSAM catalogs**
  - **Separation of application data sets and catalogs**
  - **Dedication of volumes to application data**
  - **If this scheme is followed, easy implementation**



VSE/VSAM

VSE/VSAM Backup/Restore

Compaction in VSE/VSAM 2.2 (VSE/ESA 1.3)

- **Software compaction of Backup data is possible**
- **Uses more CPU time**
- **Saves tape (or now DASD) space for backup**
- **Redundant when tape provides IDRC and when using VSE/VSAM compression (hardware or software)**

VSE/VSAM 6.1 (VSE/ESA 2.1)

- **Compression of VSAM data on disk possible**
- **BACKUP will copy compressed data to tape without expansion/compression required**
- **Can be done together with IDRC**

VSE/VSAM Backup/Restore...

Performance Aspects

- **The usual suspects**
  - **Tape speed is usually NOT the culprit**
  - **Tape channel speed is usually not the culprit**
  - **CPU speed is usually not the culprit**
  - **Disk channel speed is usually not the culprit**
  - **Disk read speed almost always is the culprit**
- **How to reduce the impact?**
  - **Specify BLOCKSIZE(65535) (the maximum)**
  - **Specify BUFFERS(6) or (8) (pre-queues I/O requests for disk and tape)**
  - **`"// UPSI 1"` for page fixing sometimes helps**
  - **Multiple concurrent backups are very effective**

VSE/VSAM Backup/Restore...

Backup to disk enhancement

- **Operational backups can be done disk to disk**
  - **Avoids tape mounting for operational backups**
  - **Enhanced unattended operations**
- **Performance very sensitive to BLOCKSIZE**
  - **Experiment with different values to find optimum**
  - **COMPACT will save disk space on backup volume**
With COMPACT, consider no BUFFERS(...) spec.

Other Hints

- **On tapes, IDRC better than COMPACT**
- **If VSAM COMPRESSED or IDRC tapes, do not use COMPACT**

VSE/VSAM Large KSDS Support

VSAM Data Set size limit -- 4 GB (VSE/ESA 2.3)

- **CIs addressed by 4-byte RBAs**
  2^32-1 = 4 GB, or a bit more than 4 Billion
- **Becoming a very significant limitation**
- **VSE/VSAM 6.3 (in VSE/ESA 2.3) adds**
  - **KSDS support for files greater than 4 GB**
  - **Internal pointers changed from RBA to CI number**
    - **These pointers not visible to applications**
- **Implementation**
  - **New DEFINE CLUSTER attribute, XXL.**
  - **Not available for other than KSDS**
- **(DL/I 1.11 adds "Data Set Groups"**
  to extend size of databases -- up to 5 ESDS = 20 GB db size)

Summary

Knowledge of VSAM Internals is critical

- **To design efficient applications and data sets**
- **To implement high performance workloads**
- **To improve performance**
- **To recover from failure**
- **To diagnose problems**

Summary...

Share with programmers and users

- **To sets expectations**
- **To provide a sound basis for performance factors in design phase**
- **To improve overall system throughput by avoiding inefficient practices**

Summary

Keeping up to date with VSAM Functions and Maintenance

- **To permit efficiency in programming**
- **To avoid integrity exposures**
- **To enable performance features**
  - **VSE/ESA extended address spaces**
  - **VSE/ESA virtual disks**
  - **New hardware developments**
  - **CICS/VSE and CICS Transaction Server enhancements in diagnosis and performance**
  - **VSE/VSAM LSR Buffer Pool Hashed Search**

Summary...

VSE/VSAM Development Continues

- **To enhancing performance**
- **To improving reliability**

Check with IBM Software Support for recent maintenance

End