1.

Record Nr.	UNINA9910824526203321
Titolo	VSAM demystified / / [Dave Lovelace ... et al.]
Pubbl/distr/stampa	San Jose, CA, : IBM, International Technical Support Organization, 2003
Edizione	[2nd ed.]
Descrizione fisica	xxiv, 494 p. : ill
Collana	IBM redbooks
Altri autori (Persone)	LovelaceDave
Disciplina	006.8
Soggetti	Virtual computer systems Digital computer simulation
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"September 2003."
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover -- Contents -- Figures -- Tables -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Chapter 1. VSAM basics -- 1.1 A brief description of VSAM -- 1.2 VSAM functions by release level -- 1.3 What is VSAM? -- 1.3.1 VSAM access types -- 1.4 Major VSAM parts -- 1.4.1 Catalog management -- 1.4.2 Record management -- 1.5 VSAM terminology and concepts -- 1.5.1 Logical record -- 1.5.2 Key field -- 1.5.3 Ways to identify logical records -- 1.5.4 Physical record -- 1.5.5 Control interval -- 1.5.6 Control area -- 1.5.7 Spanned records -- 1.5.8 Component -- 1.5.9 Cluster -- 1.5.10 Sphere -- 1.5.11 Alternate indexes -- 1.5.12 Splits -- 1.5.13 VSAM buffering -- 1.6 VSAM data set organizations -- 1.6.1 Key-sequenced data set -- 1.6.2 Entry sequenced data set (ESDS) -- 1.6.3 Relative record data set -- 1.6.4 Variable relative record data set -- 1.6.5 Linear data set (LDS) -- 1.7 Comparing VSAM data set organizations -- 1.8 Choosing a VSAM data set type -- 1.9 Extended format data set -- 1.10 Extended addressability -- 1.11 Data striping -- 1.12 Processing a VSAM cluster -- 1.12.1 Allocating a VSAM cluster -- 1.12.2 Accessing VSAM cluster -- 1.12.3 Unallocation -- 1.13 VSAM exploiters -- 1.13.1 DB2 -- 1.13.2 Hierarchical file system (HFS) -- 1.13.3 zSeries File System (zFS) -- 1.13.4 CICS -- 1.13.5 DFSMShsm -- 1.13.6 DFSMSrmm -- 1.13.7 Java Record I/O (JRIO) -- Chapter 2. Performance -- 2.1 Service level agreement -- 2.2 Transaction performance -- 2.3 Performance management -- 2.3.1 I/O performance -- 2.4 VSAM performance

management -- 2.5 VSAM rule-of-thumb mode -- 2.5.1 Invalid rules-of-thumb -- 2.6 Parameters affecting performance -- 2.6.1 Allocation units -- 2.6.2 Guaranteed Space -- 2.6.3 Optimizing control area (CA) size -- 2.6.4 Partial release.

2.6.5 Allocation constraint relief -- 2.6.6 Control interval size -- 2.6.7 FREESPACE definition for KSDS and ESDS -- 2.6.8 Index options -- 2.6.9 Key Range and Ordered -- 2.6.10 Share options -- 2.6.11 Initial load option -- 2.6.12 Region size -- 2.6.13 Buffering options -- 2.6.14 Buffering techniques -- 2.6.15 Data compression -- 2.6.16 VSAM Data striping -- 2.7 VSAM performance by scenarios -- 2.7.1 Performance scenario using RMF reports -- 2.7.2 Reducing the number of I/Os -- 2.7.3 I/O wait time (IOSQ) for VSAM data sets -- 2.7.4 I/O service time (connect) for VSAM data sets -- 2.7.5 Decreasing VSAM CPU time -- 2.8 VSAM and ESS controllers -- 2.8.1 ESS model 800 enhancements -- 2.8.2 Lab experiments -- 2.9 Performance monitors -- 2.9.1 Resource measurement facility -- 2.9.2 Tivoli Decision Support (TDS) -- 2.9.3 Generalized Trace Facility (GTF) -- Chapter 3. VSAM problem determination and recovery -- 3.1 VSAM problem determination hints and tips -- 3.1.1 How to check your VSAM data set -- 3.1.2 z/OS system messages -- 3.1.3 Catalog Search Interface IGGCSIVS program -- 3.1.4 System LOGREC messages -- 3.1.5 GTF CCW traces -- 3.1.6 DITTO/ESA output -- 3.1.7 What can you get from the SMF records? -- 3.2 Some common VSAM problems -- 3.2.1 Lack of virtual storage -- 3.2.2 Initial loading problems -- 3.2.3 Mismatch between catalog and data set -- 3.2.4 Hardware errors -- 3.2.5 Bad data or bad channel program -- 3.2.6 Structural damage -- 3.2.7 Improper sharing -- 3.2.8 Mismatch between catalog and VTOC -- 3.2.9 VSAM does not produce expected output -- 3.2.10 VSAM RLS problems -- 3.2.11 VSAM and DFSMStvs considerations -- 3.2.12 OEM problems -- 3.2.13 Enqueue issues -- 3.2.14 Migration issues -- 3.2.15 Performance considerations -- 3.2.16 Deadlocks -- 3.2.17 Beware of some VSAM restrictions -- 3.3 What documentation to collect.

3.3.1 Catalog performance problems -- 3.3.2 VSAM RLS problems -- 3.3.3 IDCAMS problems -- 3.3.4 Broken VSAM data set -- 3.3.5 Broken catalog -- 3.3.6 How to obtain VSAM record management trace? -- 3.4 How to recover a damaged VSAM data set -- 3.4.1 EXAMINE command -- 3.4.2 DIAGNOSE command -- 3.4.3 VERIFY command -- 3.4.4 Broken Index scenario -- 3.4.5 Abend task scenario -- 3.4.6 Recovering damaged BCS entries -- 3.4.7 Recovering damaged VVDS entries -- 3.5 Prevention is better than cure -- 3.5.1 Back up your VSAM data sets -- 3.5.2 Keep your system at current maintenance levels -- 3.5.3 Use Resource Recovery Management Services (RRMS) -- 3.6 Where to look for more information -- 3.6.1 IBM manuals and sources of relevant information -- 3.6.2 Information APARs from IBMLINK on VSAM problems -- 3.6.3 Information APARs on specific problems -- 3.6.4 VSAM information on the Internet -- 3.7 IDC3009I message -- 3.8 IDCAMS LISTCAT output fields -- 3.8.1 High used RBA value (HURBA) for KSDS -- 3.8.2 High allocated RBA value (HARBA) -- 3.8.3 FREESPC -- 3.8.4 High key RBA/CI -- 3.8.5 High-level index RBA value -- 3.8.6 Sequence set first RBA value -- 3.8.7 Number of index levels -- 3.8.8 Time stamps -- 3.9 SMF record types related to VSAM data sets -- 3.9.1 SMF record type 60 -- 3.9.2 SMF record type 61 -- 3.9.3 SMF record type 62 -- 3.9.4 SMF record type 63 -- 3.9.5 SMF record type 64 -- 3.9.6 SMF record type 65 -- 3.9.7 SMF record type 66 -- 3.9.8 SMF record type 67 -- 3.9.9 SMF record type 68 -- 3.9.10 SMF record type 69 -- 3.9.11 SMF record type 42 -- 3.10 RRMS and VSAM -- Chapter 4. Managing your VSAM data sets -- 4.1

Reorganization considerations -- 4.1.1 CI/CA splits -- 4.1.2 The loss of useful space in Data CA -- 4.1.3 CI/CA splits causing free space increase -- 4.2 New Index CI size calculation algorithm -- 4.2.1 Analyze existing data sets.

4.3 Sharing VSAM data sets -- 4.3.1 Write and read integrity -- 4.3.2 VSAM sharing mechanisms -- 4.3.3 Sharing data in a single VSAM control block structure -- 4.3.4 Sharing data with many VSAM control block structures -- 4.3.5 General share options: Considerations -- 4.3.6 Protecting VSAM data set through DISP parameter -- 4.4 Extended addressability (EA) -- 4.5 Catalog Search Interface -- 4.5.1 CSI setup -- 4.6 Major sources of VSAM processing options -- 4.6.1 ACB control block -- 4.6.2 DD statement keywords -- 4.6.3 Catalog BCS and VVDS entries -- 4.6.4 SMS constructs -- 4.7 Media Manager, Open, Close, EOV in VSAM -- 4.7.1 OPEN macro -- 4.7.2 CLOSE macro -- 4.7.3 End-of-Volume (EOV) -- 4.8 VSAM and 64 bits -- 4.9 Special considerations for COBOL users and SMB -- 4.9.1 COBOL users take note -- Chapter 5. VSAM Record Level Sharing -- 5.1 Introducing VSAM RLS -- 5.1.1 What is VSAM RLS? -- 5.1.2 Why RLS? -- 5.1.3 How does RLS work? -- 5.1.4 RLS in a single system (monoplex) -- 5.1.5 CICS and VSAM RLS -- 5.1.6 RLS restrictions -- 5.2 RLS terminology -- 5.3 Planning for RLS -- 5.3.1 Hardware requirements -- 5.4 Implementing VSAM RLS -- 5.4.1 Define Sharing Control Data Set (SHCDS) -- 5.4.2 Define CF cache structures -- 5.4.3 Define CF lock structures -- 5.4.4 SMS definitions -- 5.4.5 Modifying the PARMLIB IGDSMSxx Member -- 5.4.6 Security definitions -- 5.4.7 Using a VSAM sphere in RLS mode -- 5.4.8 RLS Recoverable spheres -- 5.5 RLS problem determination and recovery -- 5.5.1 Problems with SHCDS -- 5.5.2 Problems with SMSVSAM -- 5.5.3 Problems with locks -- 5.5.4 SHCDS FALLBACK procedure -- 5.5.5 RLS rules -- 5.5.6 MVS commands for RLS -- 5.6 RLS enhancements -- 5.6.1 RLS/KSDS extended addressability -- 5.6.2 VSAM RLS CF structure duplexing and rebuild -- 5.6.3 RLS CF caching enhancements -- 5.7 RLS performance.

5.7.1 Factors affecting RLS data sharing performance -- 5.7.2 CF access time -- 5.7.3 RLS performance gains -- 5.7.4 CICSPlex RLS performance comparison -- 5.7.5 Batch RLS performance experiments and comparison -- 5.7.6 RMF and VSAM RLS -- 5.7.7 MVS commands about RLS performance -- 5.7.8 SMF records covering VSAM RLS -- Chapter 6. DFSMStvs -- 6.1 Introducing DFSMStvs -- 6.2 Why DFSMStvs? -- 6.2.1 How to extend CICS availability -- 6.2.2 Reducing the batch window -- 6.3 Some definitions -- 6.3.1 Backward recovery -- 6.3.2 Forward recovery -- 6.3.3 Atomic updates -- 6.3.4 Unit of work and unit of recovery -- 6.3.5 Two-phase commit -- 6.3.6 In-flight and in-doubt -- 6.3.7 Repeatable read -- 6.3.8 Recoverable data sets -- 6.4 CICS support for recoverable VSAM -- 6.5 DFSMStvs overview -- 6.5.1 The RLS connection -- 6.5.2 DFSMStvs locking -- 6.5.3 DFSMStvs logging -- 6.5.4 Recovery coordination -- 6.6 Our experiences with implementation -- 6.6.1 Define list structures in the CFRM policy -- 6.6.2 Define the log structures and log streams in LOGR policy -- 6.6.3 Define SMS constructs for DFSMStvs -- 6.7 DFSMStvs problem determination tips -- 6.7.1 How to take a dump of the problem? -- 6.7.2 Classes of errors -- 6.7.3 Determining the Failing Module -- 6.7.4 Apparent batch job hangs -- 6.7.5 Other hangs -- 6.7.6 Quiescing a data set -- 6.7.7 Close/delete/rename of data set with inflight UR -- 6.7.8 New and changed system level commands for DFSMStvs -- 6.7.9 SET SMS and SETSMS commands -- 6.7.10 VARY SMS command -- 6.7.11 SYS1.PARMLIB changes -- 6.7.12 Changes to Job Control Language (JCL) -- 6.7.13 Changes to IDCAMS -- 6.7.14 Messages and codes -- 6.7.15 Macros that have been changed to

support DFSMStvs -- Appendix A. Sample code -- JRIO API examples -- Locate a record by key in keyed access record file -- Position to a record in a random access record file.
Read a record from a keyed access record file.

Sommario/riassunto

Virtual Storage Access Method (VSAM) is one of the access methods used to process data. Many of us have used VSAM and work with VSAM data sets daily, but exactly how it works and why we use it instead of another access method is a mystery. This book helps to demystify VSAM and gives you the information necessary to understand, evaluate, and use VSAM properly. It clarifies VSAM functions for application programmers who work with VSAM. The practical, straightforward approach should dispel much of the complexity associated with VSAM. Wherever possible an example is used to reinforce a description of a VSAM function. This IBM Redbook is intended as a supplement to existing product manuals. It is intended to be used as an initial point of reference for VSAM functions. This book also builds upon the subject of Record Level Sharing and the new z/OS feature called DFSMStvs.