UNIVERSITÄT LEIPZIG

Enterprise Computing
Einführung in das Betriebssystem z/OS

Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

WS2012/2013

Datasets Teil 2

VSAM Struktur

VSAM

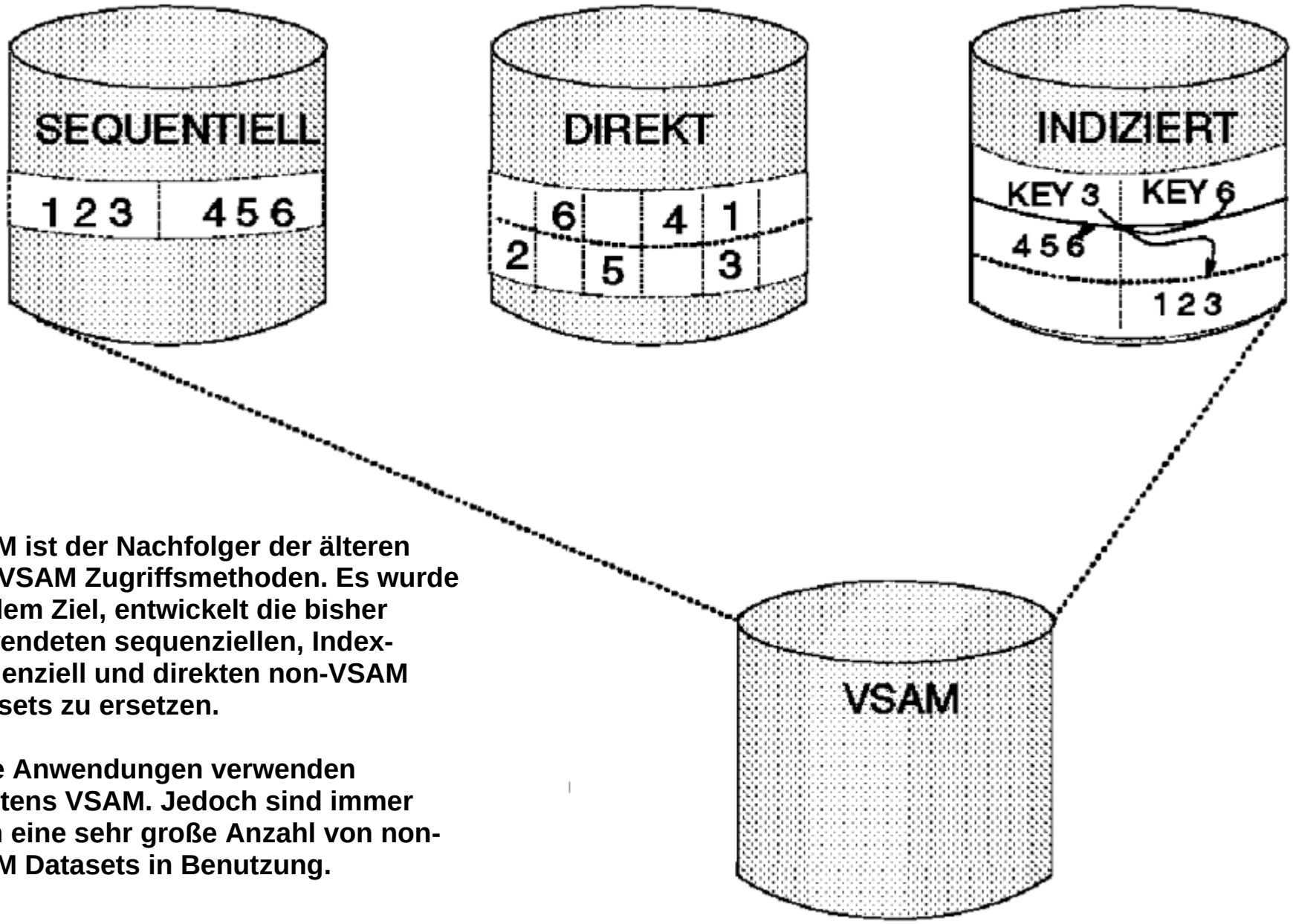
VSAM (Virtuell Storage Access Method) ist die wichtigste z/OS-Zugriffsmethode. Sie wurde speziell für das Arbeiten in einer z/OS virtual Storage Umgebung entwickelt.

Der Begriff Virtual Storage Access Method (VSAM) beschreibt sowohl einen Datensatz-Typ (Organisation) als auch die Zugriffsmethode, mit der auf verschiedene Anwender-Datentypen zugegriffen wird. Als Zugriffs Methode stellt VSAM weit komplexere komplexe Funktionen zur Verfügung als andere bisherige Zugriffs-Methoden. VSAM speichert Daten in einem speziellen Format auf der Festplatte ab, das nicht für andere Zugriffsmethoden verständlich ist.

VSAM wird hauptsächlich für die Speicherung von Anwendungsdaten verwendet. In vielen Fällen wird VSAM in Situationen eingesetzt, in denen unter Linux oder Windows der Einsatz einer relationalen Datenbank erforderlich wäre. VSAM ist nicht für Source-Programme, JCL, oder ausführbare Module vorgesehen. VSAM Datasets können nicht routinemäßig mit ISPF angezeigt oder bearbeitet werden.

Sie können VSAM benutzen, um Datensätze in vier Arten von Datensätzen zu organisieren: entry-sequenced, key-sequenced, relative record (direct), oder linear. Der Hauptunterschied zwischen diesen Typen von Datensätzen ist die Art, wie Records gespeichert und abgerufen werden.

VSAM-Datensätze können mit einer z/OS-System Utility namens IDCAMS erstellt werden.



VSAM ist der Nachfolger der älteren non-VSAM Zugriffsmethoden. Es wurde mit dem Ziel, entwickelt die bisher verwendeten sequenziellen, Index-sequenziell und direkten non-VSAM Datasets zu ersetzen.

Neue Anwendungen verwenden meistens VSAM. Jedoch sind immer noch eine sehr große Anzahl von non-VSAM Datasets in Benutzung.

VSAM Data Sets

Es existieren die folgenden Arten von VSAM Data Sets:

Entry Sequence Data Set (ESDS)

Diese Form von VSAM speichert Records in sequentieller Reihenfolge. Datensätze können sequentiell abgerufen werden. ESDS wird auch als Container verwendet, um ein vollständiges z/OS Hierarchical File System zu speichern.

Key Sequence Data Set (KSDS)

Dies ist die häufigste Verwendung von VSAM. Jeder Datensatz enthält ein (oder mehrere) Schlüsselfelder. Ein Datensatz kann über seinen Schlüssel gelesen (oder insertiert) werden. Dies ermöglicht einen Zugriff über einen Index auf die Daten. Die Records können von unterschiedlicher Länge sein.

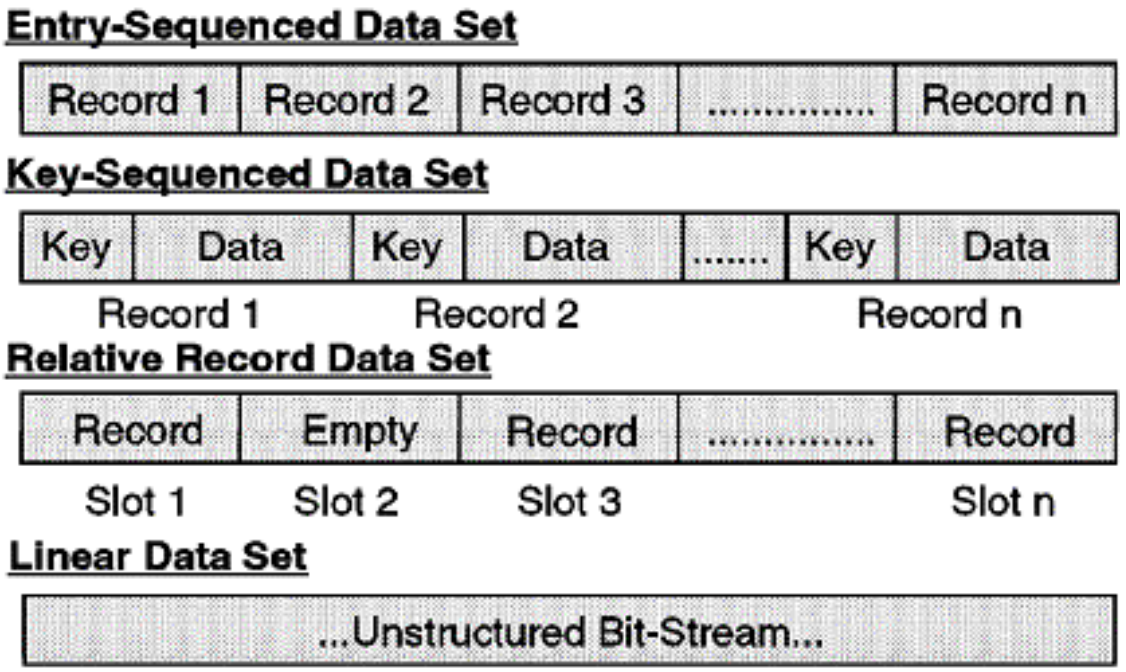
Relative Record Data Set (RRDS)

Dieses VSAM-Format ermöglicht das Abrufen von Datensätzen nach ihrer Nummerierung; Datensatz 1, Satz 2, und so weiter. Dies ermöglicht einen random (direkten) Zugang zu den Daten, vorausgesetzt das Anwendungsprogramm kennt die Nummer des Records. Das Anwendungsprogramm muss eine Möglichkeit haben, die gewünschte Datensatz-Nummer zu ermitteln.

In den meisten RRDS Datasets haben alle Records die gleiche Länge. Ein VRRDS (variable relative Rekord Dataset) ermöglicht Records mit unterschiedlicher Länge.

Linear Data Set (LDS)

Dies ist ein Byte-Stream Datensatz (ähnlich einer Unix-Datei). Eine beträchtliche Anzahl von z/OS-System Funktionen benutzen dieses Format; es wird aber nur selten von Anwendungsprogrammen verwendet.



VSAM Data Set Types

Ein Relative Record Data Set wird auch als Direct Record bezeichnet.

ESDS Datasets sind für die sequentielle Verarbeitung von Daten optimiert.

Benutzung von VSAM

Die VSAM Zugriffsmethode (Access Method) dient als Schnittstelle zwischen Anwendungsprogrammen und dem Betriebssystem. Ein Anwendungsprogramm ruft VSAM Zugriffsmethode Routinen als normale Unterprogramme auf.

In Assembler Language-Programmen erfolgt die Benutzung durch einen Aufruf von VSAM-Makros. Bei der Benutzung von Hochsprachen wie Cobol, C++ oder PL/1 wandelt der Compiler I/O-Anweisungen (z.B. WRITE) in Aufrufe an die entsprechenden VSAM-Routinen um. Wenn die I/O-Anforderung abgearbeitet wurde, wird die Steuerung an das Anwendungsprogramm zurückgegeben.

Beim Zugriff auf einen Record durch ein Anwendungsprogramm geht VSAM durch die folgenden Schritte:

1. **VSAM interpretiert den Aufruf des Anwendungsprogramms und bestimmt, welche Dienste erwünscht sind.**
2. **VSAM erstellt die erforderlichen Input-oder Output (I/O) request(s) an das Betriebssystem.**
3. **Das Betriebssystem führt die physischen I/O-Operation(en) zwischen Festplatte und Hauptspeicher durch.**
4. **VSAM lokalisiert und extrahiert die gewünschten Daten zur Rückgabe an das Anwendungsprogramm.**

Zwei wichtige Alternativen beim Zugriff auf einen VSAM Record sind:

1. **Beim Zugriff auf einen Festplattenspeicher (Direct Access Storage Device, DASD) transportiert VSAM (bzw. z/OS) einen größeren Block (Control Interval) mit mehreren Records vom/zum virtuellen Hauptspeicher. Der vom Anwendungsprogramm gewünschte Record befindet sich möglicherweise in einem Control Intervall, welches bereits in den virtuellen Speicher geladen wurde. Eine physische I/O-Operationen ist in diesem Fall nicht erforderlich.**
2. **Aufgrund der Art wie VSAM Daten speichert und der Vielfalt der Verarbeitungsoptionen kann es sein, dass mehrere physische I/O-erforderlich sind, um einen einzigen logischen Record in den virtuellen Speicher zu laden.**

z/OS Catalog

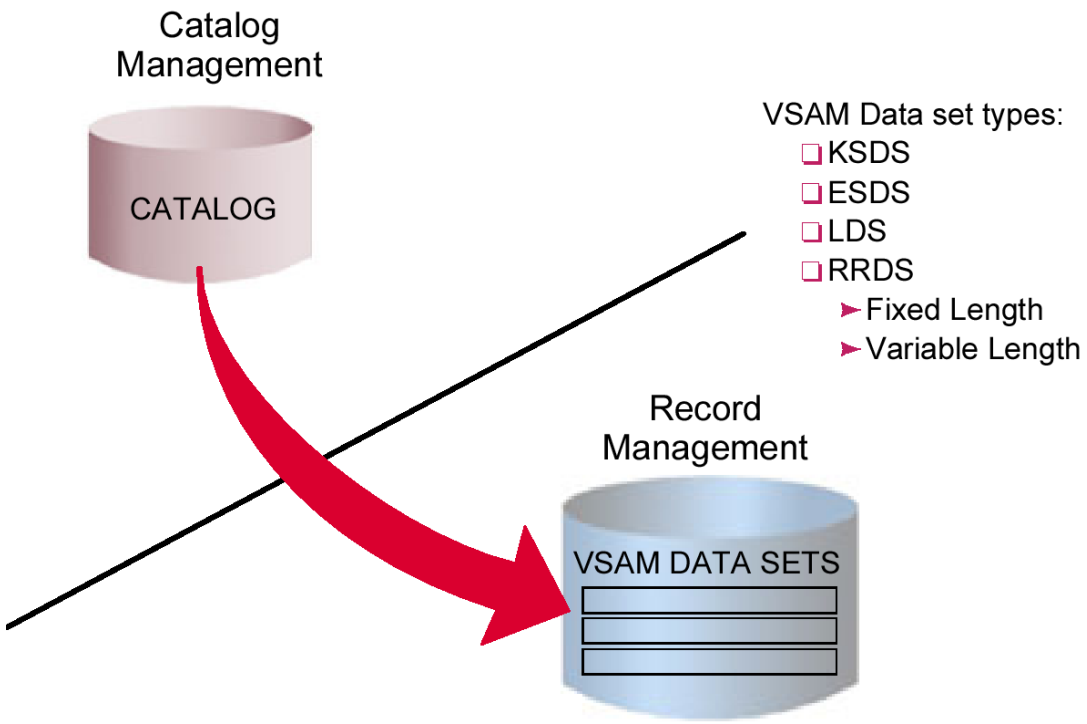
Unter Windows kann eine Datei wahlweise in der Partition C:, D:, E: bis Z: gespeichert werden. Der Benutzer muss wissen, wo die Datei gespeichert ist, wenn er darauf zugreifen will. Auch können 2 verschiedene Dateien unter dem gleichen Namen in zwei verschiedenen Partitionen gespeichert werden.

Unter z/OS hat jeder Dataset einen eindeutigen Namen, meistens beginnend mit der Benutzer-ID. Zum Beispiel könnte der Benutzer prak123 einen Record in einem Dataset mit dem Namen PRAK123.TEST01.COBOL speichern.

Eine z/OS-Struktur, der „Katalog", ist ein systemweites Verzeichnis, in dem der Speicherort aller Datasets (VSAM, non-VSAM, PDSE) enthalten ist. Er enthält Informationen, auf welchem Plattenlaufwerk der Dataset PRAK123.TEST01.COBOL gespeichert ist. Bedenken Sie, eine z/OS-Installation kann viele Tausende von Festplatten enthalten. Unterschiedliche Arten von Datasets (VSAM, non-VSAM, PDS) können katalogisiert werden. Werden auf einem Windows Rechner unterschiedliche File Systeme (z.B. NTFS, FAT32) in unterschiedlichen Regions eingesetzt, ist dies nicht ohne weiteres möglich.

In der Praxis werden z/OS Datasets fast immer katalogisiert.

Für Details siehe IBM Redbook: “VSAM Demystified”. September 2003, SG24-6105-01
http://www.redbooks.ibm.com/redbooks/pdfs/sg246105.pdf



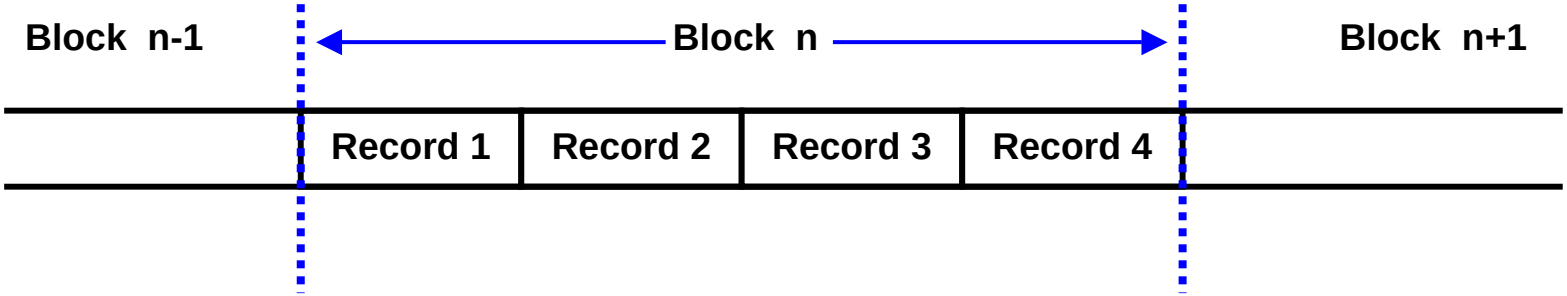
Virtual Storage Access Method (VSAM)

Die VSAM-Access Method definiert, wie VSAM Datasets in Katalogen organisiert und verwaltet werden. Darüber hinaus legt sie fest, wie Records in einer VSAM Dataset organisiert sind. Damit hat die VSAM Access Method zwei Hauptfunktionen:

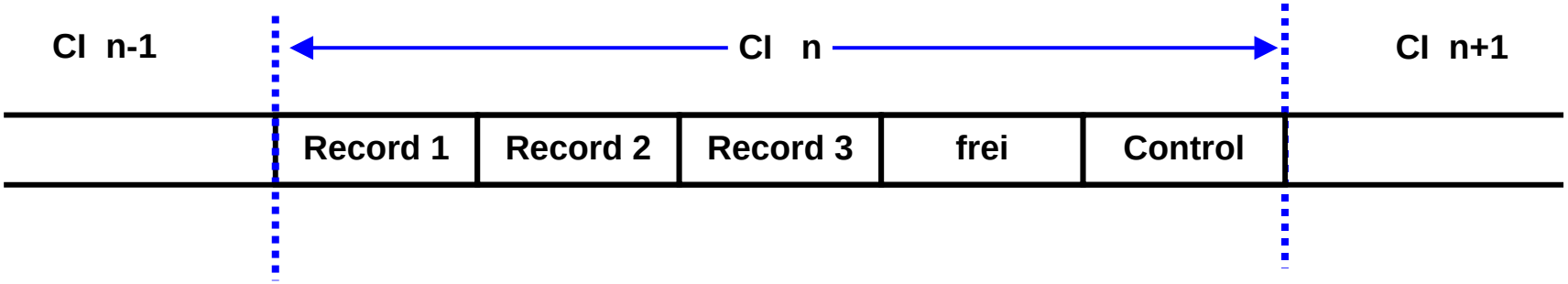
- **Das Record Management verwaltet die Records eines VSAM Datasets**
- **Das Catalog Management enthält Funktionen vergleichbar mit einem Unix File Directory, optimiert für die Verwaltung von einer wesentlich größeren Anzahl von Datasets.**

VSAM Datasets werden andersartig als non-VSAM Datasets auf dem Plattenspeicher abgespeichert..

VSAM speichert Records in Blöcken, die als Control Intervals bezeichnet werden Ein Control Interval (CI) ist ein zusammenhängender (contiguous) Bereich auf einem DASD (disk drive), welcher sowohl Records als auch Steuerinformation speichert. Daten werden zwischen Hauptspeicher und Plattenspeicher als ganze Control Intervals bewegt. Die Größe der CIs kann von einem VSAM Dataset zum nächsten VSAM Dataset unterschiedlich sein; alle CIs innerhalb eines spezifischen VSAM Datasets haben jedoch die gleiche Länge. Eine sehr häufig benutzte Control Interval Größe ist 4 KByte, identisch mit der Größe eines 4 KByte Virtual Storage Page Frames.



In einer normalen Queued Access Method (z.B. QSAM) fasst man mehrere (logische) Records zu einem Block (physischer Record) zusammen. Der physische Record ist die Dateneinheit, die zwischen Festplatte und I/O Puffer im Hauptspeicher gelesen und geschrieben wird.



Ein VSAM Control Intervall (CI) unterscheidet sich von einem Block (physischer Record dadurch, dass es neben mehreren (logischen) Records noch zusätzliche Control Information über die Records des CI speichert. Außerdem kann in einem CI freier Platz für die zukünftige Aufnahme weiterer Records vorhanden sein.

Ein Control Interval besteht aus

- **mehreren logischen Records,**
- **freien Platz, plus einem**
- **Control Interval Definition Field (CIDF) sowie mehreren**
- **Record Definition Fields (RDFs).**

In so fern unterscheidet sich ein CI von den Blöcken (physischen Records) anderer Dataset Access Methods, bei denen ein Block lediglich eine Aneinanderreihung von (logischen) Records enthält.

Mehrere Control Intervals in einem VSAM Dataset werden in einem zusammenhängender (contiguous) Bereich auf einem DASD (disk drive) zusammengefasst, der als Control Area bezeichnet wird. Eine Control Area hat häufig die Größe eines Zylinders (15 Spuren) einer 3390 Festplatte, oder 849960 Bytes.

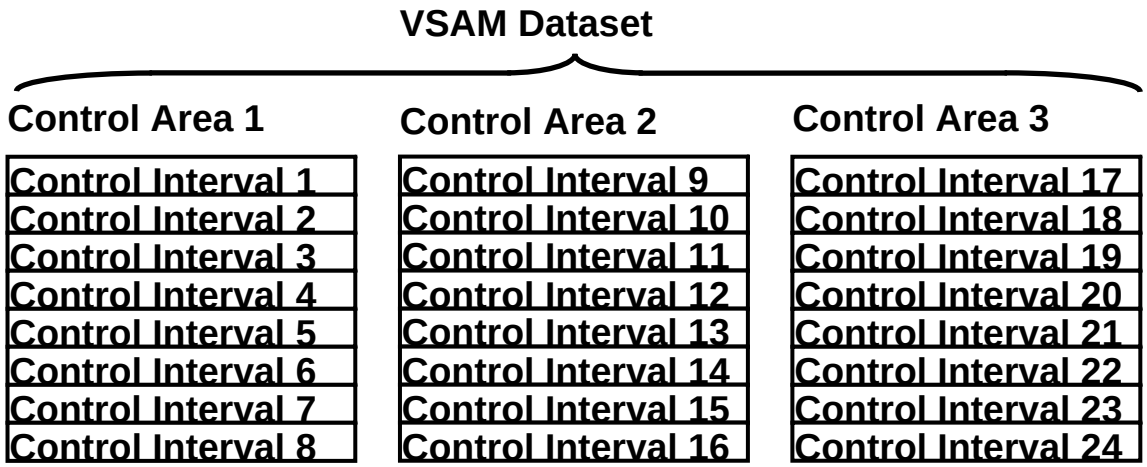


Abb.1.1 : Beispiel eines VSAM Datasets, der aus 24 Control Intervallen und 3 Control Areas besteht.

Ein VSAM Dataset kann aus vielen Control Areas bestehen. Mit einer Control Interval Größe von 4 KByte ist eine maximale Größe eines VSAM Datasets von 16 TBbyte möglich.

LR = (logischer) Record
RDF = Record Descriptor Field
CIDF = Control Interval DescriptorField

Struktur eines VSAM Control Intervals

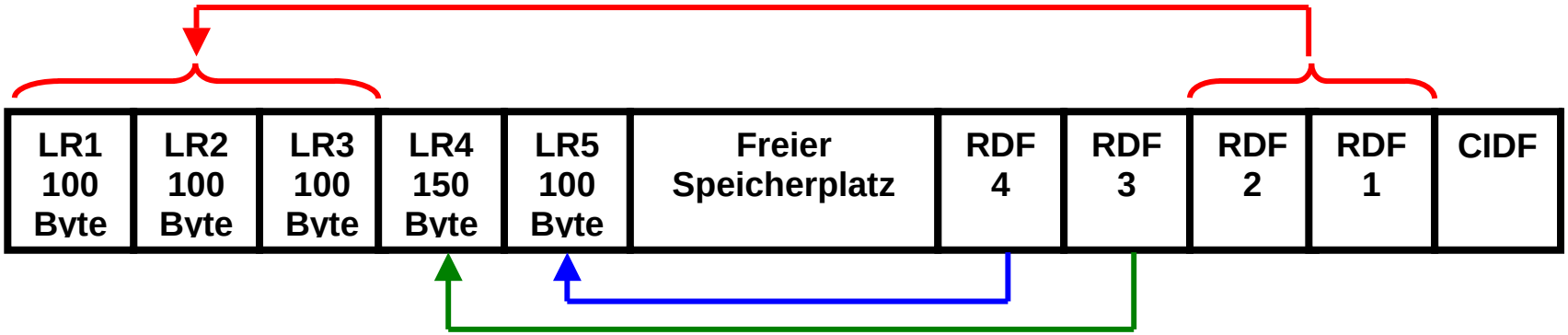
VSAM arbeitet mit Blöcken, die als „Control Intervals“ (CI) bezeichnet werden. Ein CI enthält typischerweise mehrere (logische) Records. Die Standard CI Größe ist 4KByte, kann aber bis zu 32KByte betragen.

Das Control Intervall enthält

- **· Records,**
- **· Ungenutzten (freien) Speicherplatz,**
- **· Rekord-Deskriptor Felder (RDF) und**
- **· ein einziges Control Interval Deskriptor Feld (CIDF).**

Die CIDF ist ein 4-Byte-Feld. Es enthält Informationen über die Größe und Lokation des freien Speicherplatzes. Ein RDF ist ein 3-Byte-Feld. Es beschreibt die Länge der Records. VSAM benutzt häufig Records mit variabler (unterschiedlicher) Länge. In diesem Fall existiert ein RDF für jeden Record. Für eine Folge von Records mit fester Länge benutzt man 2 RDFs. Ein RDF gibt den Wert der Länge an, der zweite RDF besagt, wie viele Records mit gleicher Länge vorhanden sind.

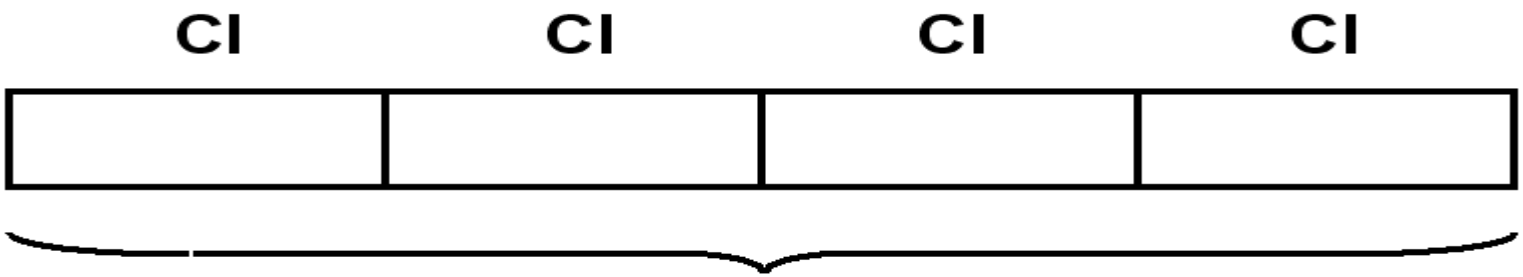
In so fern unterscheidet sich ein CI von den Blöcken (physischen Records) anderer Dataset Access Methods, bei denen ein Block lediglich eine Aneinanderreihung von (logischen) Records enthält.



Aufeinanderfolgende (contiguous) Records mit identischer Länge

Das hier gezeigte Beispiel zeigt ein Control Interval mit mehreren Records sowie frei verfügbaren Platz für die Aufnahme weiterer Records. Der Datenbereich enthält 5 logische Records mit jeweils einer Länge von 100 Bytes, 100 Bytes, 100 Bytes, 150 Bytes und 100 Bytes.

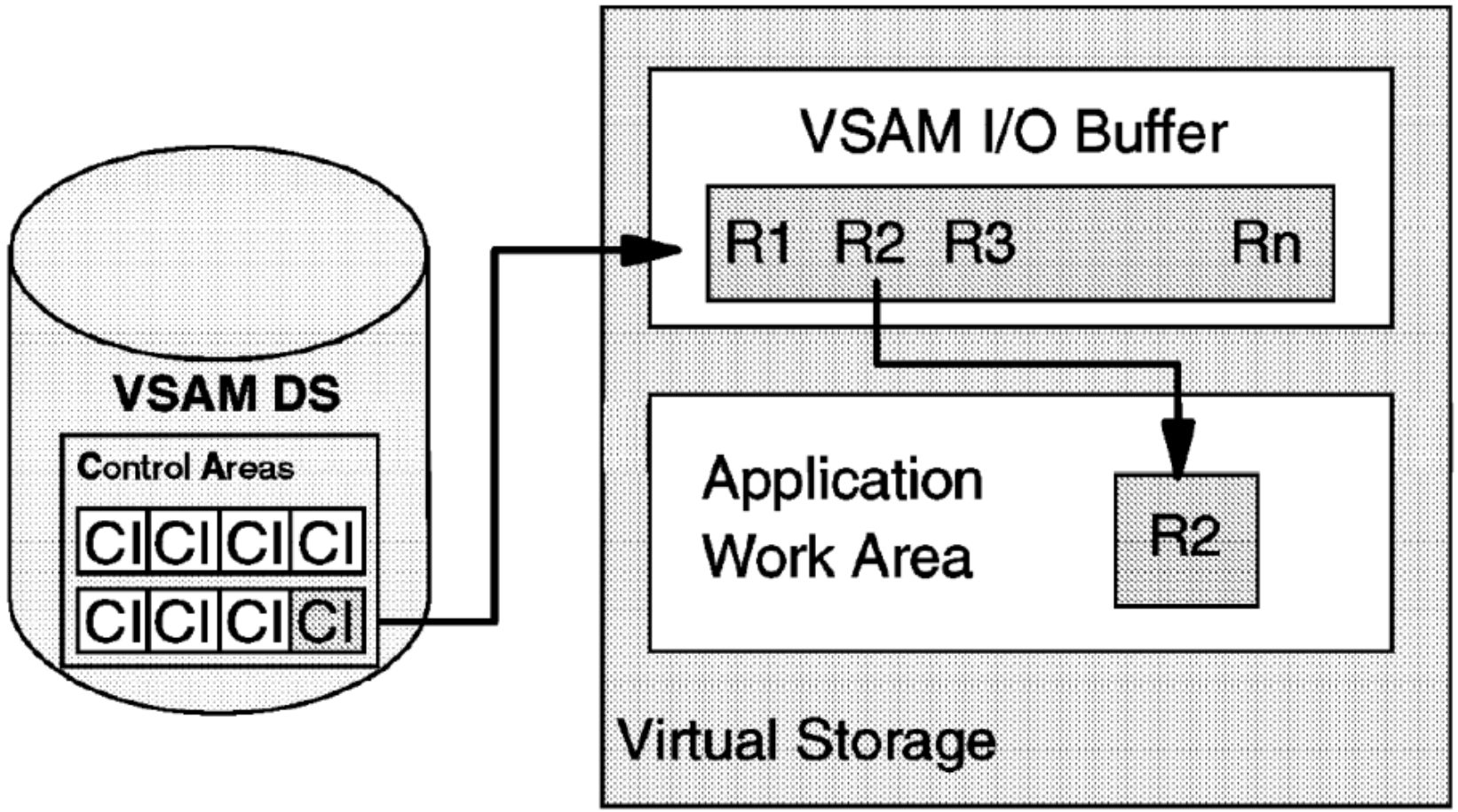
Es existiert 1 Control Interval Definition Field (CIDF), welches die Lokation und Länge des freien (nicht genutzten) Speicherplatzes angibt. Von den 4 Record Definition Fields (RDF) definieren die beiden ersten RDFs die Länge (100 Bytes) und die Anzahl (3) der ersten Gruppe von 3 Records mit je einer Länge von 100 Bytes. Die beiden nächsten RDFs definieren die Länge der folgenden beiden Records.



Control Area (CA)

Eine Control Area ist ein spezifisches VSAM Construct. Ein Control Area wird von zwei oder mehreren Control-Intervallen gebildet, und in einem zusammenhängenden (contiguous) DASD Bereich gespeichert. Ein VSAM Dataset besteht häufig aus einer größeren Anzahl von Control Areas.

In vielen Fällen hat eine Control Area die Größe eines IBM Typ 3390 Festplatten Zylinders (849 960 Bytes). Die Größe der CA wird definiert, wenn der Dataset allocated wird.



Wenn ein Anwendungsprogramm einen VSAM Record liest, wird das ganze Control Interval, das den Datensatz enthält, von dem DASD in einen VSAM I/O Buffer in den virtuellen Speicher des Anwendungsprozesses kopiert. Anschließend wird der gewünschte Record aus dem VSAM I/O-Puffer in einen separaten Puffer im Bereich des Anwendungsprogramms kopiert.

IDCAMS

Ein VSAM Data Set hat eine komplexe Struktur, und das Anlegen eines neuen Data Sets ist ebenfalls kompliziert.

Hierfür existiert eine Utility unter dem Namen IDCAMS.

IDCAMS ist ein Dienstprogramm unter z/OS zum Anlegen und Verwalten von VSAM Dateien. Mit IDCAMS können Systemspezialisten auch Kataloge administrieren.

Der Name setzt sich, wie bei IBM-Programmen üblich, aus einem Drei-Zeichen-Präfix IDC und der Abkürzung AMS (für Access Method Services) zusammen.

IDCAMS kann mittels JCL in einem Batchjob ausgeführt werden. Eine weitere Möglichkeit ist der Aufruf aus einem Anwenderprogramm. Hier ist es möglich, den Standardinput (SYSIN) und Standardoutput (SYSPRINT) statt mit Dateien mit eigenen Unterprogrammen zu behandeln.

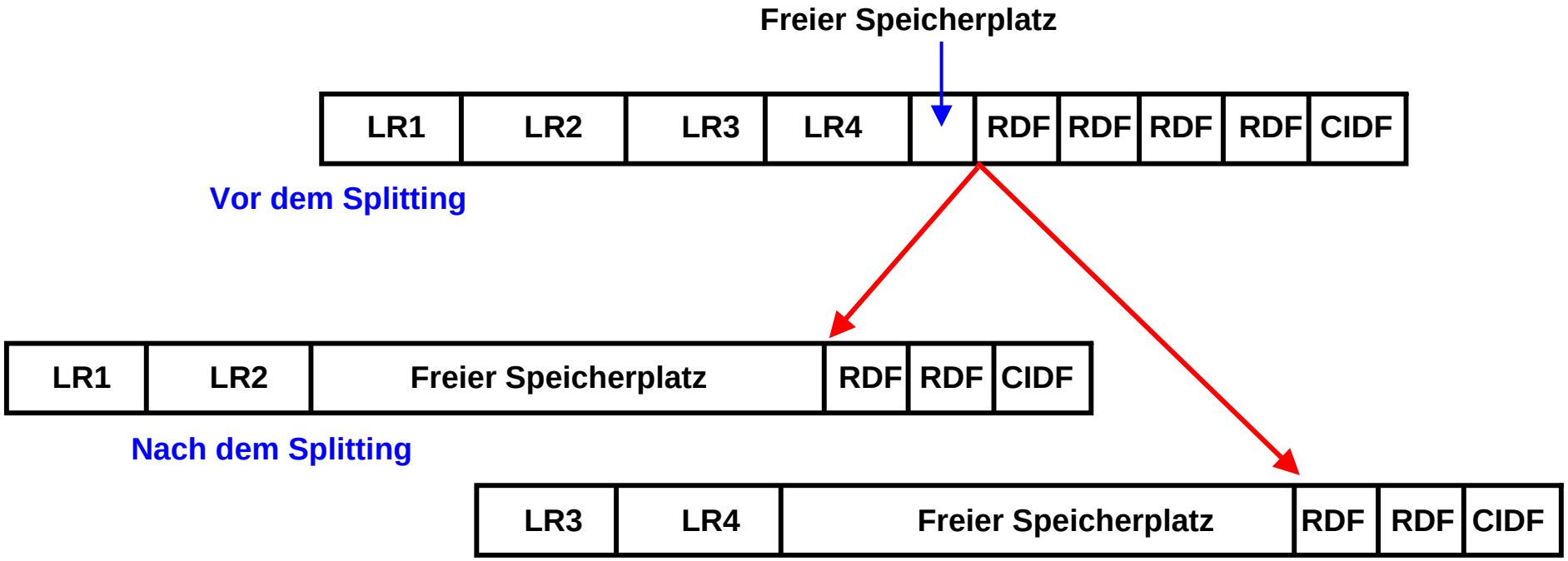
Buffering

Buffering ist einer der wichtigsten Aspekte was die I/O-Leistung betrifft. Ein VSAM Puffer ist ein virtueller Speicherbereich, in den ein CI während einer I/O-Operation übertragen wird.

Ein Bufferpool ist ein Satz von Puffern, von denen jeder ein CI aufnimmt. Ein Pufferpool kann mehrere CIs des gleichen VSAM-Dasets enthalten. Zusätzlich kann ein Anwendungsprogramm auf mehrere VSAM-Datasets zugreifen. So kann der Bufferpool häufig CIs von unterschiedlichen VSAM-Datasets enthalten.

Ein Ressourcen Pool ist ein Bufferpool mit zusätzlicher Steuerinformation. Diese beschreibt den Pool und die CIs in dem Ressourcen Pool.

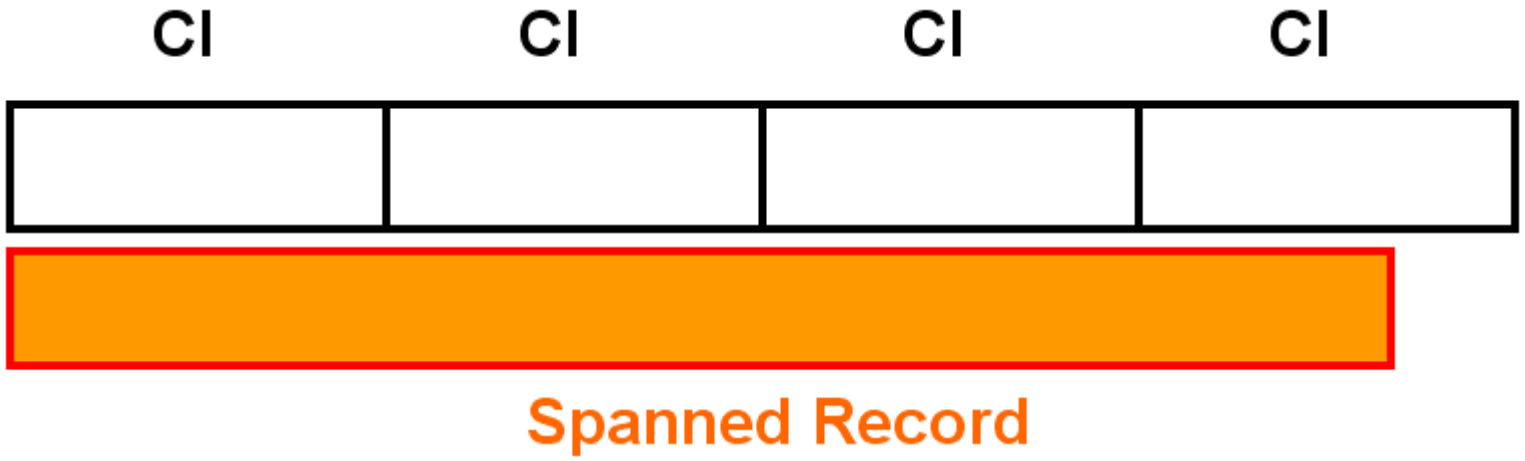
Eine Datenbank nutzt auch Bufferpools. Datenbanken wie DB2 und IMS transportieren Daten zwischen Festplatte und Hauptspeicher in Blöcken, die als „Slots" bezeichnet werden. Ein Slot hat eine Größe von 4096 Byte.



Splits

Splits treten auf, wenn der ungenutzte Speicherplatz innerhalb eines Control-Intervall ist nicht mehr groß genug ist, um einen zusätzlichen neuen Rekord aufzunehmen. In diesem Fall wird die Hälfte der Datensätze in ein neues Control Intervall bewegt. Dies stellt mindestens 50% freien Speicherplatz in jedem Control Interval bereit. Anschließend werden die Control Information Felder in jedem Control Intervall aktualisiert.

Wenn Speicherplatz in einer Control Area knapp wird, wird die Control Area ebenfalls gesplittet.



Spanned Records

Spanned Records sind logische Records, die größer als ein Control Intervall sind. Sie werden benötigt, wenn die Anwendung sehr lange logischen Records erfordert. Um mit Spanned Records zu arbeiten, muss der Dataset zum Zeitpunkt der Erstellung mit dem Spanned Records Attribut definiert werden. Spanned Records dürfen Control Interval Grenzen überschreiten. Die RDFs beschreiben, ob der Record spanned ist oder nicht.

Ein Spanned Record beginnt immer auf einer Control Intervallgrenze und füllt eine oder mehrere Control Intervalle innerhalb einer einzigen Control Area. Ein Control Interval, welches einen Teil eines Spanned Records beherbergt, enthält keine weiteren Records. In anderen Worten, der freie Platz am Ende des letzten Segments wird nicht mit dem nächsten Record gefüllt. Dieser Freiraum wird nur verwendet, um den Spanned Rekord zu verlängern.

Ein “Spanned Record” erstreckt sich über mehrere Control Intervalle, kann aber nicht größer als eine Control Area sein.