Navigating the IMS Database 4/8GB limitations

Gary Turner
IMS Software Consultant

Agenda

Why is this an issue?

→ Data Growth

→ Skills shortage

→ Understanding Database Space

Here are your alternatives

→ Do nothing, well kind of

→ HALDB - Config and migration

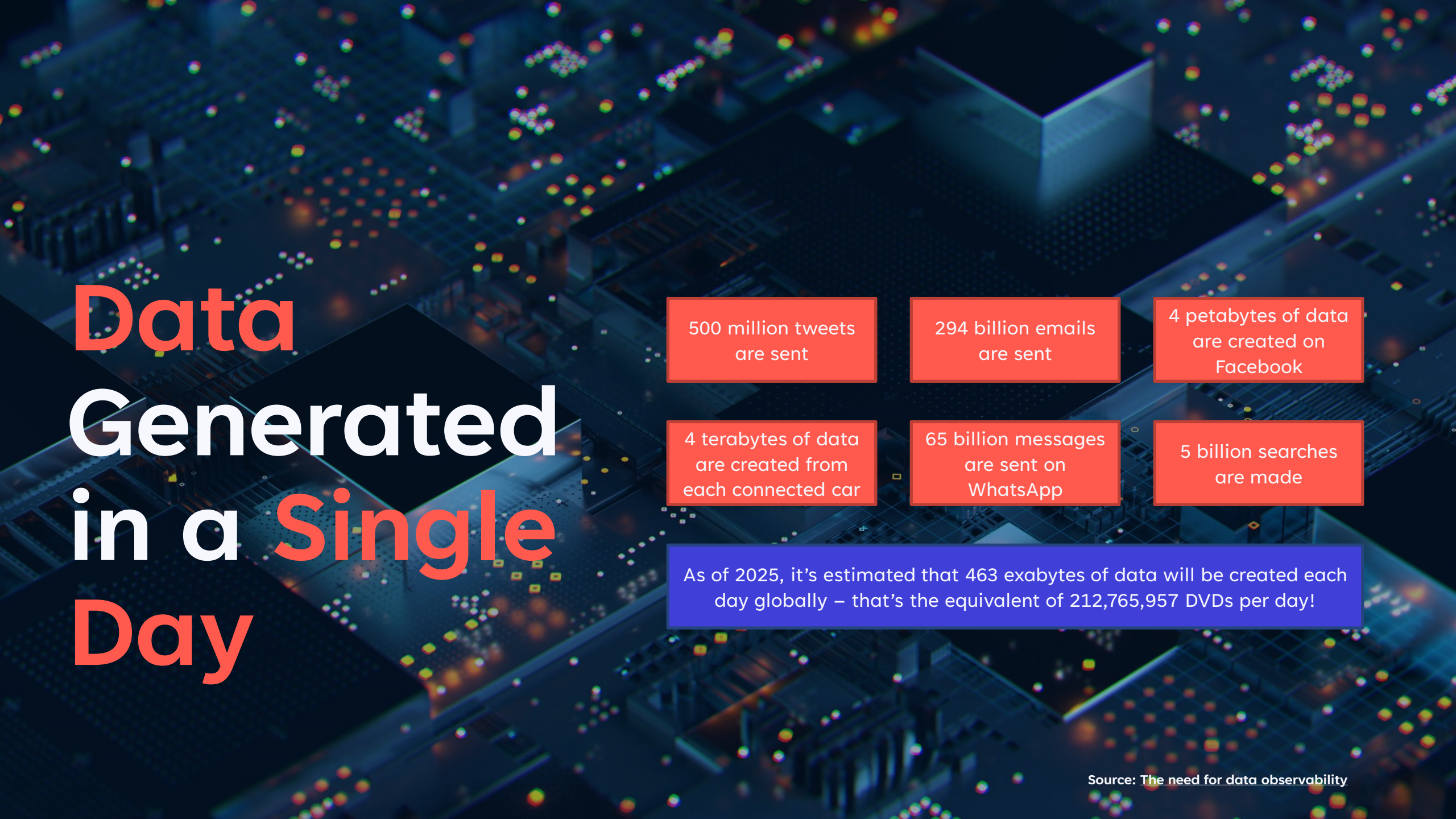
→ Fast Path - Config and migration

→ PDF - Config and migration

→ Compression - Config and migration

Conclusion

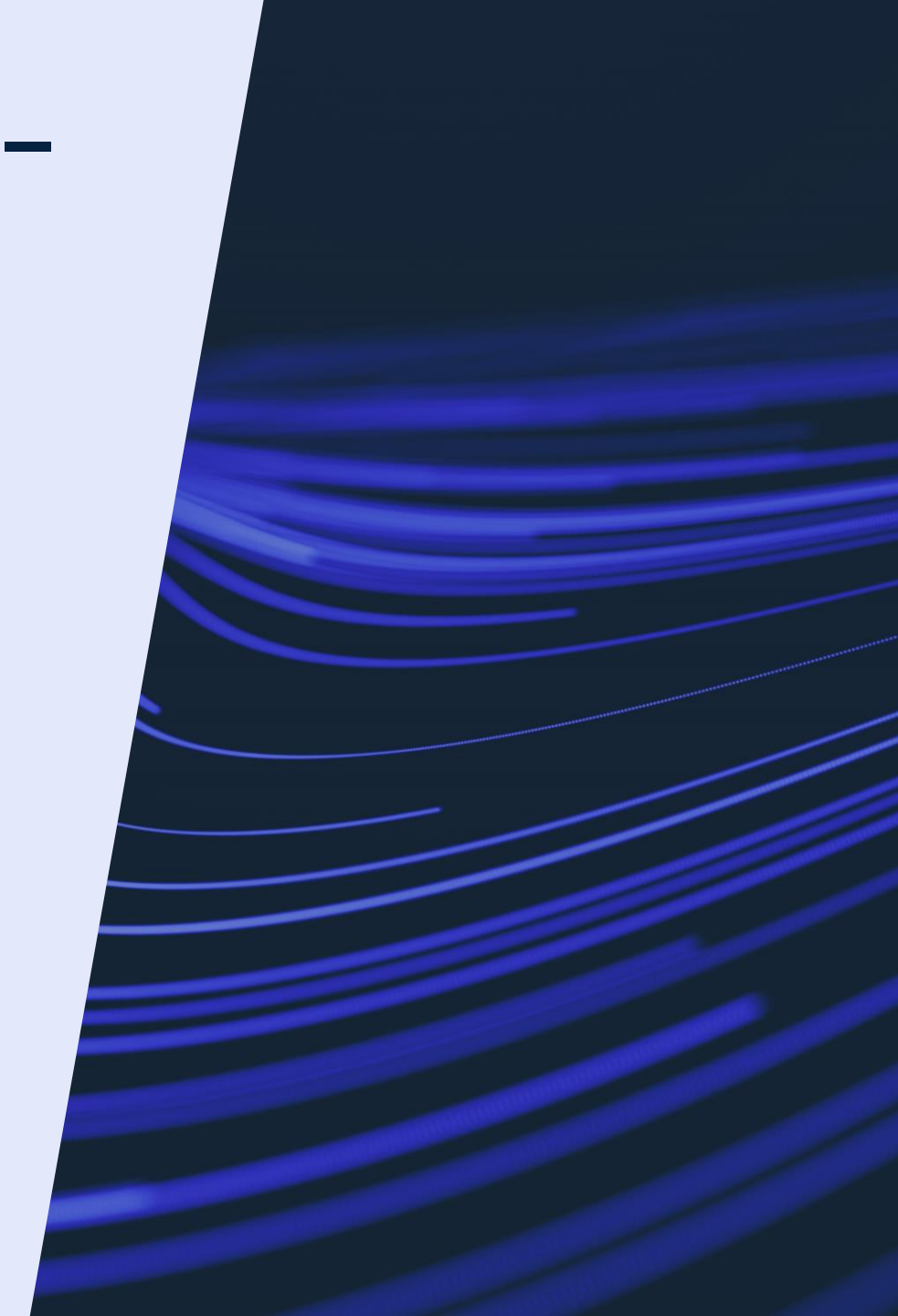
SPACE, THE FINAL FRONTIER


Data Generated in a Single Day
500 million tweets are sent
294 billion emails are sent
4 petabytes of data are created on Facebook
4 terabytes of data are created from each connected car
65 billion messages are sent on WhatsApp
5 billion searches are made
As of 2025, it's estimated that 463 exabytes of data will be created each day globally – that's the equivalent of 212,765,957 DVDs per day!
Source: The need for data observability

Example of Data Growth Issues – Stimulus Checks

Several banks across the US reported trouble with online and mobile banking Wednesday -- the same day that coronavirus IRS stimulus checks are expected to hit bank accounts of those eligible.

Banks including Chase, Capital One, PNC Bank, US Bank and Navy Federal, as well as the app Cash App, have all reported outages or access issues on Twitter support pages or their websites. Web outage monitoring site Down Detector also shows a spike in outages for these and other online banking services.



Understanding Space

A full-function database consists of one or more data sets. As IMS data increases in the data set, the free space in the data set decreases. When all the free space is used up, IMS applications cannot insert new segments or replace existing segments. To prevent this situation, you must take appropriate actions to mitigate the potential problems before they become significant issues.

Out-of-space problems can be caused by one of the following three conditions:

Primary space is used up

→ Primary allocation space is used up.

The data set is full

→ Data set size has reached the maximum limit that is defined by IMS.

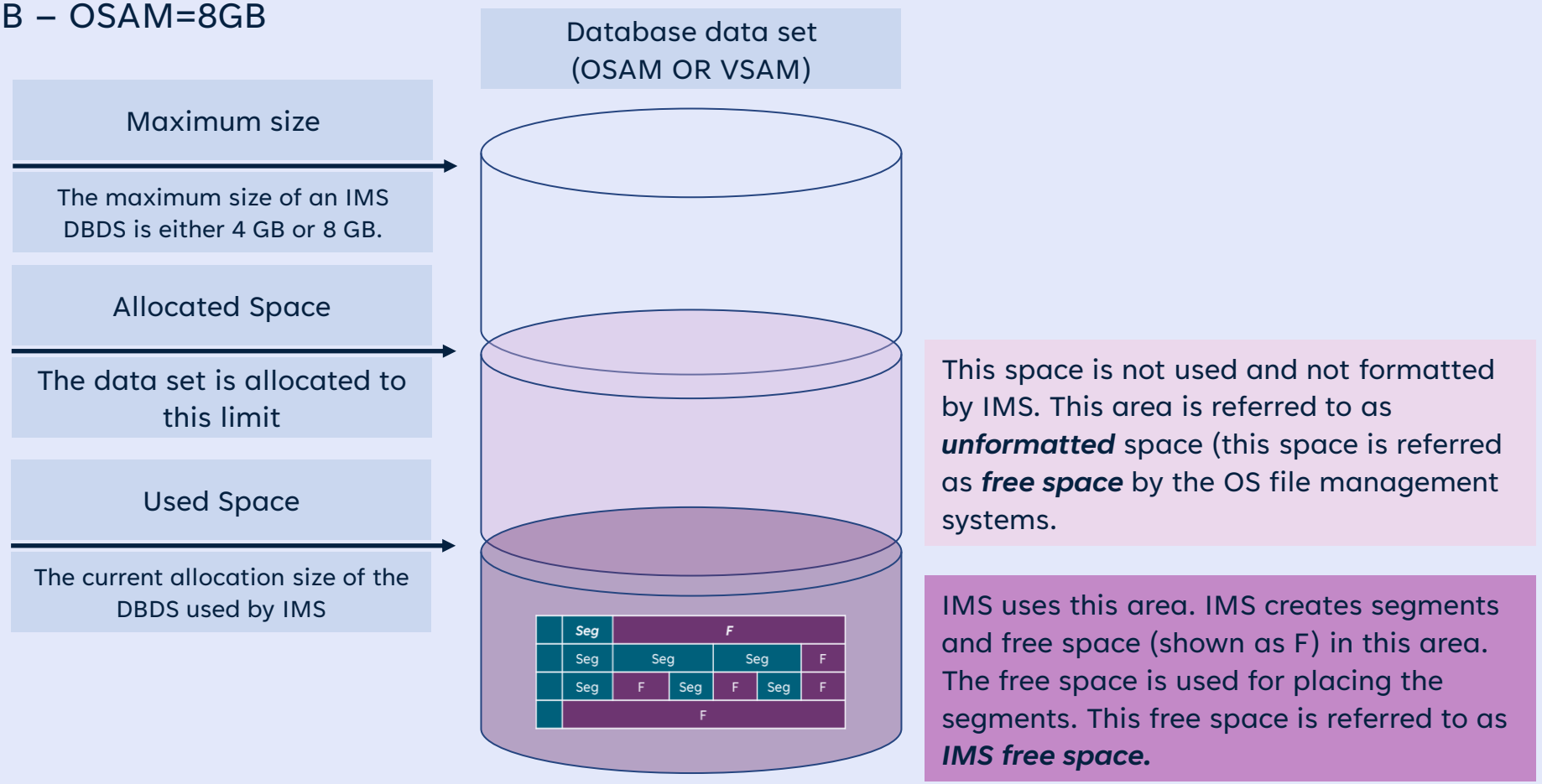
Extent resources are used up

→ Data set cannot be expanded.

How IMS uses space in datasets

The following shows how IMS uses space in a full-function database data set.

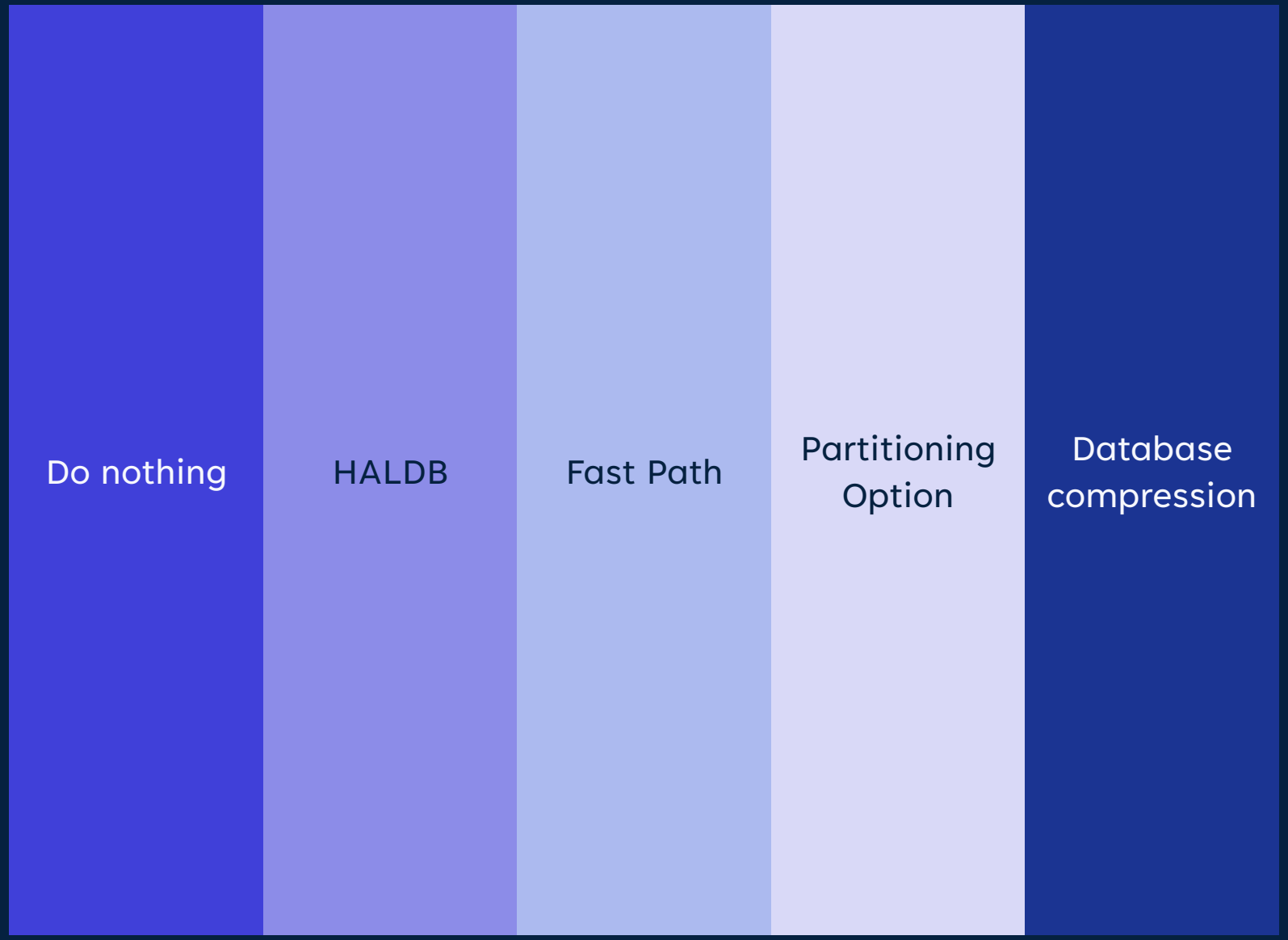
VSAM = 4GB – OSAM=8GB



This space is not used and not formatted by IMS. This area is referred to as ***unformatted*** space (this space is referred as ***free space*** by the OS file management systems.

IMS uses this area. IMS creates segments and free space (shown as F) in this area. The free space is used for placing the segments. This free space is referred to as ***IMS free space.***

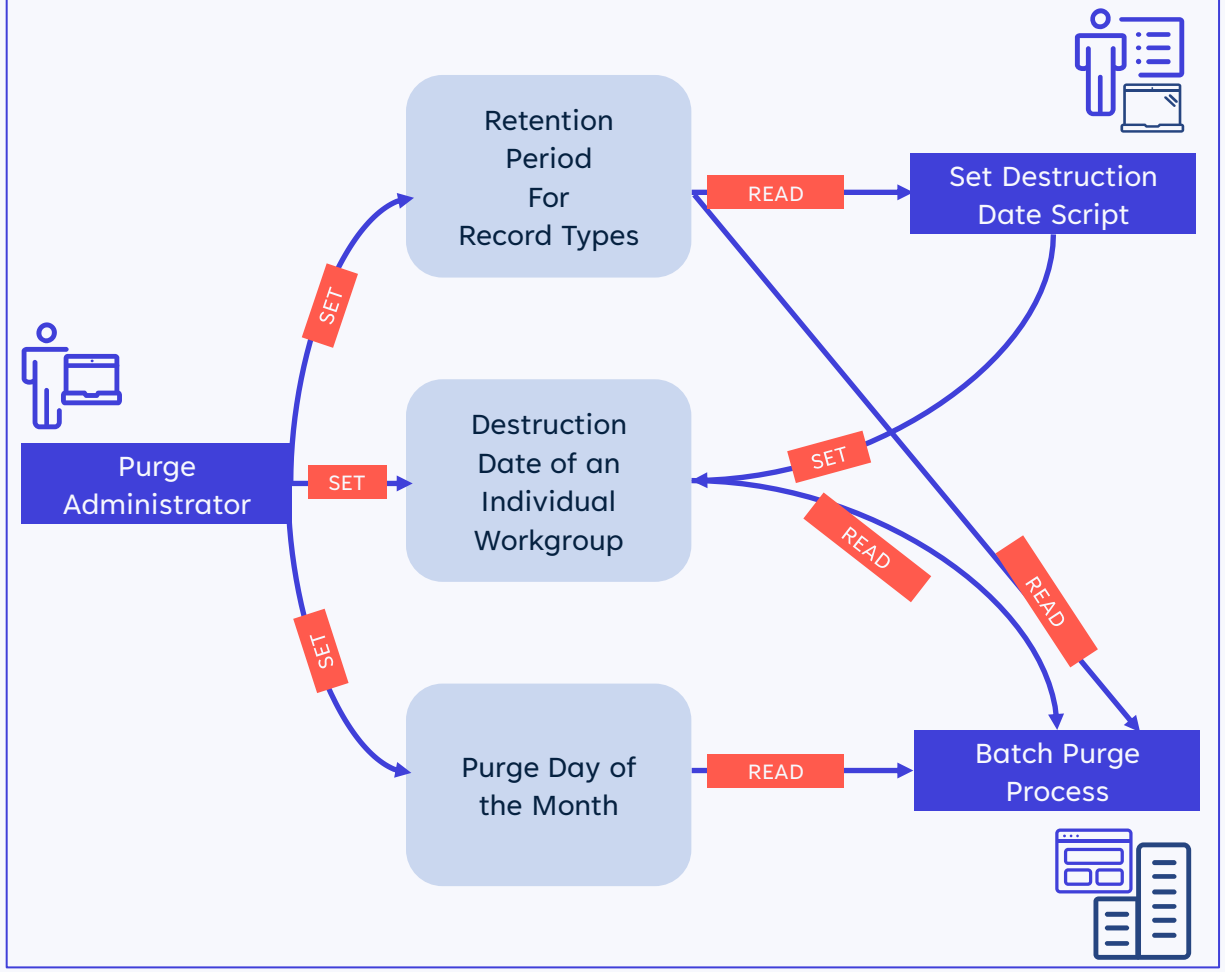
What are the alternatives



Do nothing

Do nothing?

Data Flow of the three core players and their data interaction.



DO IT YOURSELF? ?

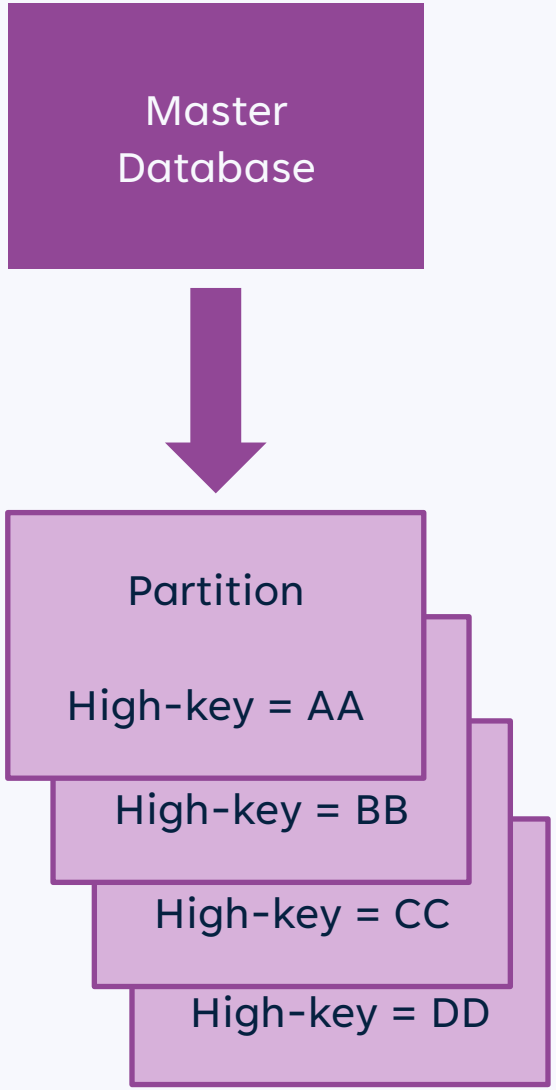
HALDB

Partitioning Options

HALDB

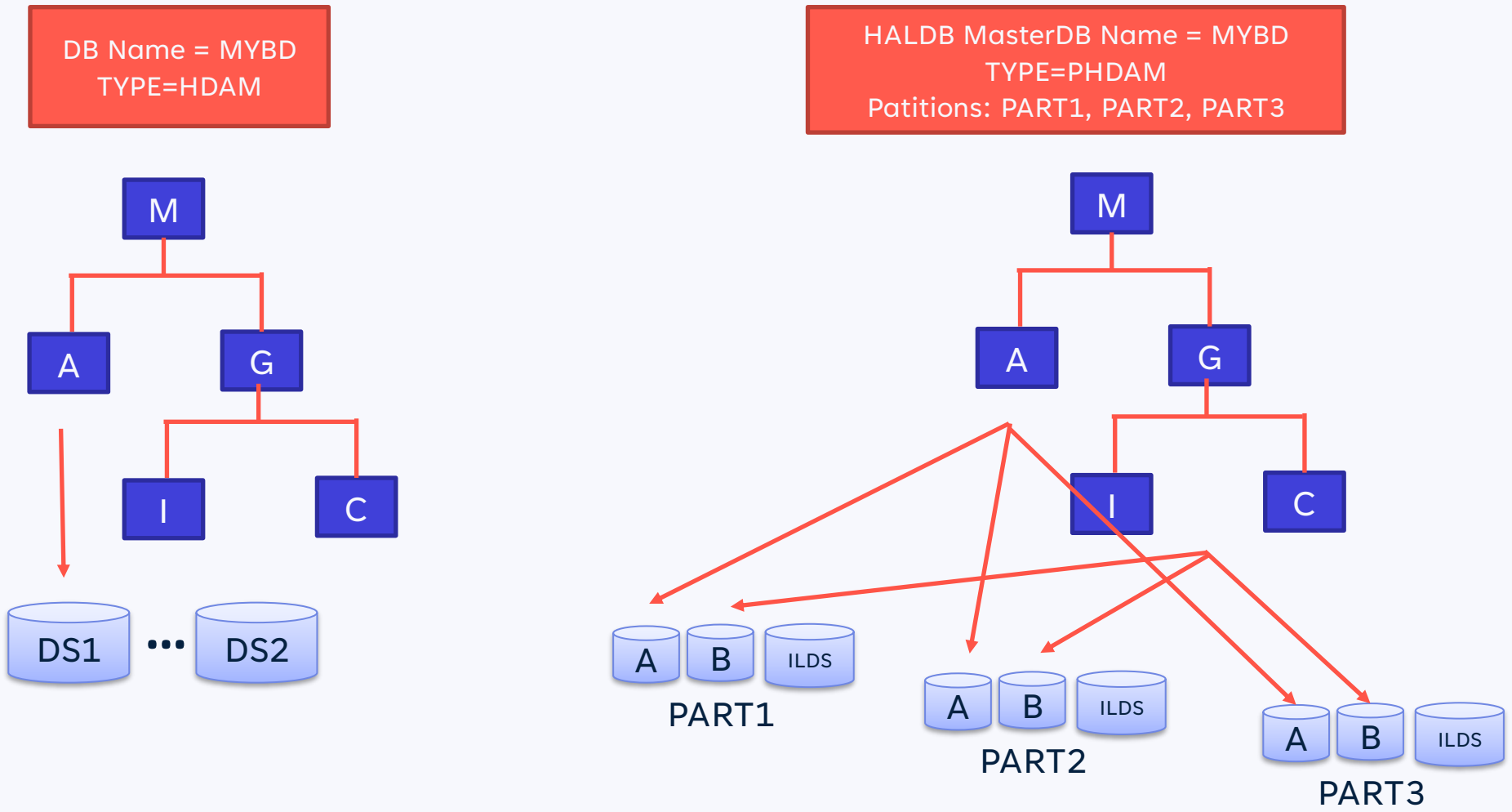
- Introduced in IMS v7 – 1997 (GA in October 2000)
- Supports up to 1001+ partitions
  - Up to 10 Datasets (max 10,010)
  - Up to 40 TB (VSAM) 80 TB (OSAM)
- Partition size max of 4 GB for VSAM and 8 GB for OSAM
- Requires DBRC

Basic Design




```
DBD NAME=MYDB,...
SEGM
NAME=SEG1,...
FIELD NAME=FIELD1,...
FIELD NAME=FIELD1,...
```

Basic Design



HALDB Migration Process

→ Use a HALDB Migration Tool

→ Unload the full function database.

→ Save DBRC information for the full function database.

→ Delete full function database information.

→ Define the HALDB information to the RECON.

→ Initialize the partitions.

→ Load the HALDB.

→ Image copy the partitions.

Note: All logically related databases and their secondary indexes must be migrated to HALDB at the same time!

Also assumes that the database name remains the same.

HALDB Migration Tool

- → Analyzes Full Function Databases
- → Helps decide partition high keys based on either
  - → Number of desired partitions
  - → Desired size of each partition
  - → Count of records
  - → Subset of root key
- → Capability to merge multiple datasets
- → Can factor in a user specified growth percentage
- → Generates DBRC statements for HALDB definition
- → Generates IDCAMs statements for HALDB partitions
- → GUI-based workflow

HALDB Migration Process

- → Use a HALDB Migration Tool
- → Image Copy the full function database.
- → Unload the full function database.
- → Save DBRC information for the full function database. (BACKUP.RECON)
- → Delete full function database information. (DELETE.DB)
- → Define the HALDB information to the RECON, ACB/Catalog, Directory.
- → Initialize the partitions. (INIT.DB & INIT.PART)
- → Load the HALDB.
- → Run INDEX build utilities
- → Image copy the partitions.

Note: All logically related databases and their secondary indexes must be migrated to HALDB at the same time!

Also assumes that the database name remains the same.

HALDB – What changes to expect

→ Database size grows after converting to HALDB – especially for secondary indexes

- Pointers grow from 4 to 28 bytes (EPS)
- RKSIZE (root Key size) keyword adds the root key to each secondary index record
- /SX (Secondary Index) fields grow from 4 to 8 bytes
- ILK (Indirect List Key - 8 bytes) assigned to each segment

→ Primary Index database is substituted by a dataset (one for each partition)

→ New dataset (Indirect List Dataset) (one for each partition)

→ Things not available in the HALDB world

- Shared secondary indexes
- Non unique keys
- Symbolic pointers
- Virtual logical relationships

HALDB – Pro's and Con's

PROs

- Storage capacity in increased dramatically
- Self Healing Pointers eliminates the need to rebuild secondary index pointers and logical relationships after a reorganization
- Partitions can be managed independently
- Parallel processing
- Transparent to Applications

CONs

- Complexity of migration
- HALDB requires DBRC in every IMS environment
- Logical relationships (Every DB must be converted)
- Secondary indexes must be PSINDEX
- Additional data sets
- Rebalancing partitions

Fast Path

Fast Path Options

Fast path - Term used in computer science to describe a path with shorter instruction path length through a program compared to the 'normal' path.

→ Introduced in IMS 1.1.4 GA 1977

→ Supported 2048 areas 8+ TB – Now it supports 9999 areas ~40 TB

→ Area size max of 4 GB

→ DBRC not required, but RECOMMENDED

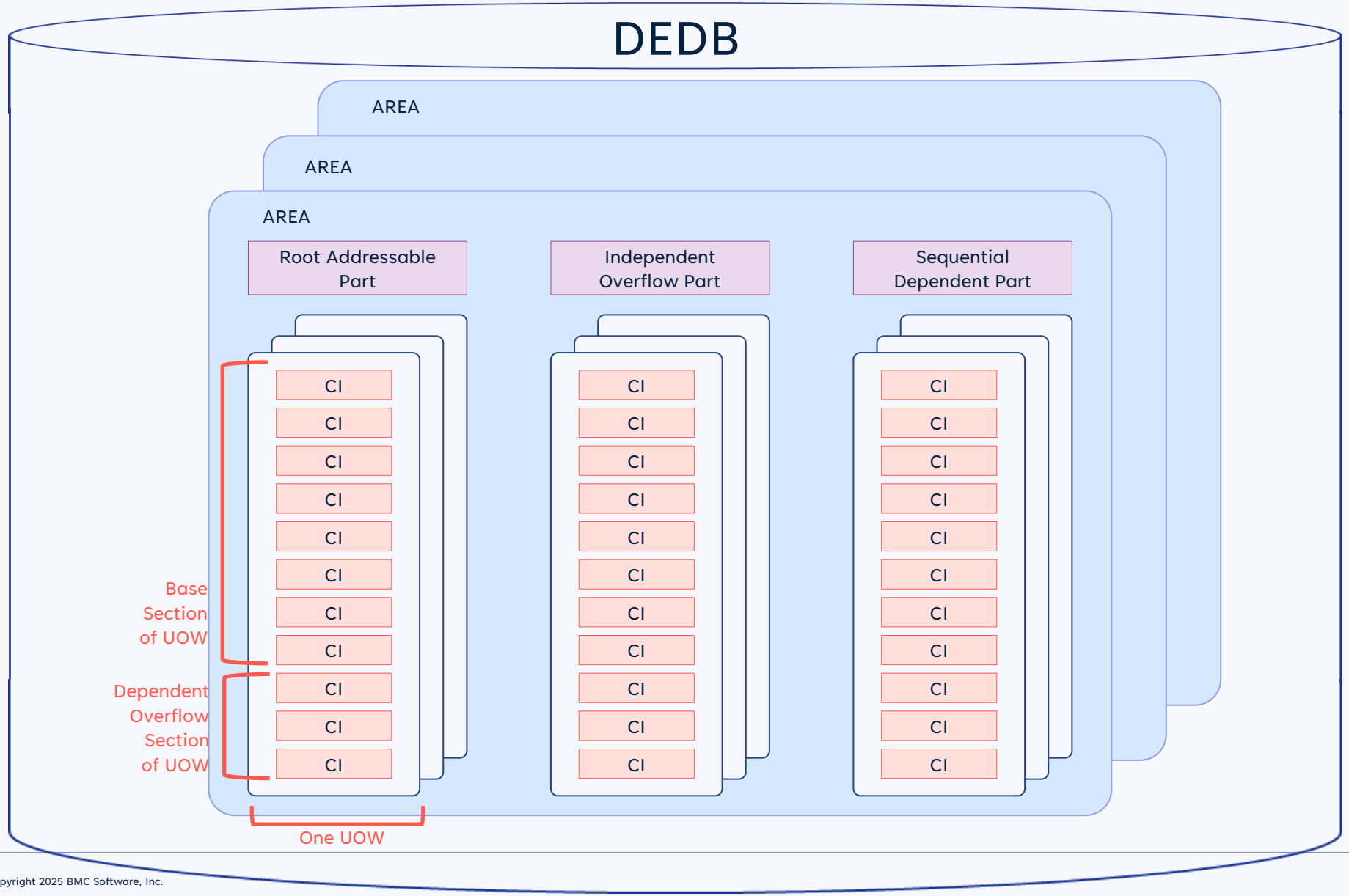
→ Support full online reorganization with concurrent update capability. No outage is required unless you are doing restructuring. DEDBs were designed with online reorganization in mind.

DEDBs Achieve High Performance

The performance-related aspects of DEDB are these:

- → Path length
- → I/O elimination and parallelism
- → Sequential dependent segments (SDEPs)
- → High-speed sequential processing(HSSP)
- → Subset pointers
- → Virtual Storage option (VSO)
- → Main Storage Databases
- → Expedited Message Handler
- → DEDB Areas / Randomizer
- → Log reduction

Parts of a DEDB area in storage



Designing a DEDB

1. Determine database structure needed for application
2. Picking a CI Size
3. Picking a Unit of Work Size
4. Designing an Area
5. Defining Your DEDB to DBRC
6. Initializing a DEDB
7. Your DEDB is now ready for you to add the data. You do not need a load mode PSB to add data to a DEDB.

When to Use HALDB/FF Databases vs DEDBs

- → Logical Relationships and *Secondary Indexes
- → Batch processing
- → Fast Path Database Buffer Pool Saturation
- → Lower transaction volumes
- → OSAM
- → Need more than 127 segment types
- → Space or performance, which is valued higher?

Partitioning Option

Partitioning Options

Partitioned Database Facility for IMS (PDF):

→ Delivered to market in 1997

- IMS/ESA® Partition Support Product (PDB)
  - created by Neon

→ Supports 127 partitions (VSAM 500GB, OSAM 1TB)

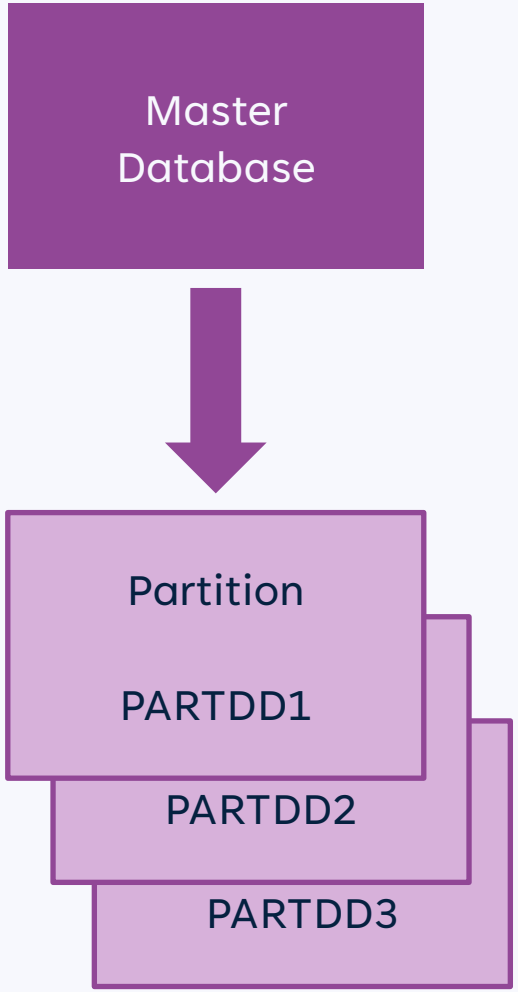
→ Includes support for HISAM, SHISAM and OSAM 8GB

→ DBRC not required

→ NO application change

→ Full support for IMS Catalog and managed ACBs

Basic Design



DBD NAME=PARTDBD,ACCESS=HIDAM,

PSNAME=PDFPSEL

or

PART DD1=PARTDD1,....

PART DD1=PARTDD2....

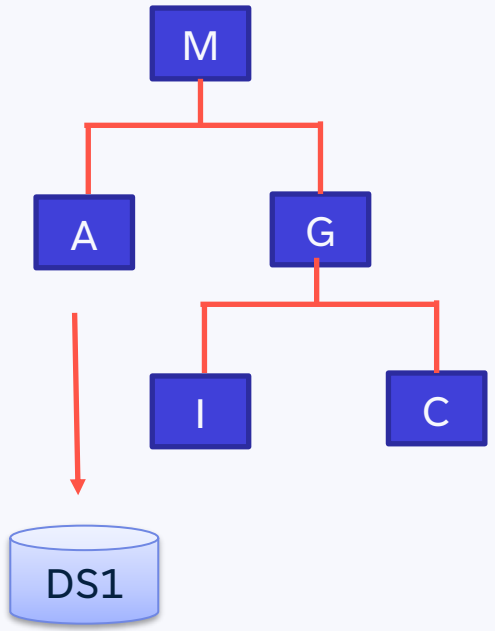
PART DD1=PARTDD3,...

DATESET(s)

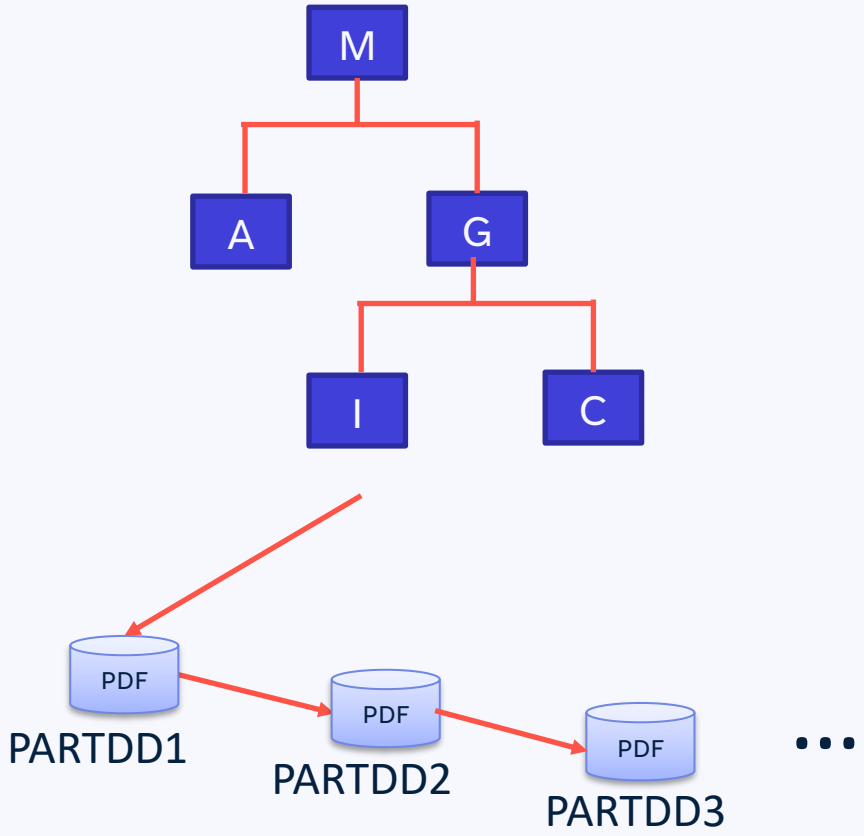
Attributes set by Part Selection Routine (PSR), register in the Recon

Basic Design

DB Name = MYBD
TYPE=HDAM



DB Name = MYBD TYPE=HIDAM
Partitions: PARTDD1, PARTDD2, PARTDD3



PDF Migration

1. Unload database (a single HD unload file)
2. Add PART statement(s) to DBD
3. Run DBD and ACB GENs for new DBD or New Catalog/Directory GENs
4. Notify DBRC (optional)
5. Update MDALIB
6. Load partitioned database
7. Build secondary index(es)
8. Image Copy database
9. Alter any JCL

PDF – Pro's and Con's

PROs

→ Significantly expands database size by distributing records across multiple partitions

→ Transparent to applications and PSBs

→ No performance hit; can improve performance

- Reduced contention
- Improves utility performance due to parallelism
- Allows for parallel batch processes

→ May improve database manageability and recoverability

→ Supports partitioning of indexes as well

CONs

PDF Summary

Simple, straightforward solution that minimizes complexity, risks, and costs

→ Business as usual

- Program testing environment and processes are unchanged
- Operationally consistent – little or no change to production procedures

→ Minimal learning curve

→ Installs quickly

- Nominal effort to migrate
  - Very easy to implement DBD changes
  - No application changes
  - No PSB changes

→ Testing process for applications is unchanged

Database Compression

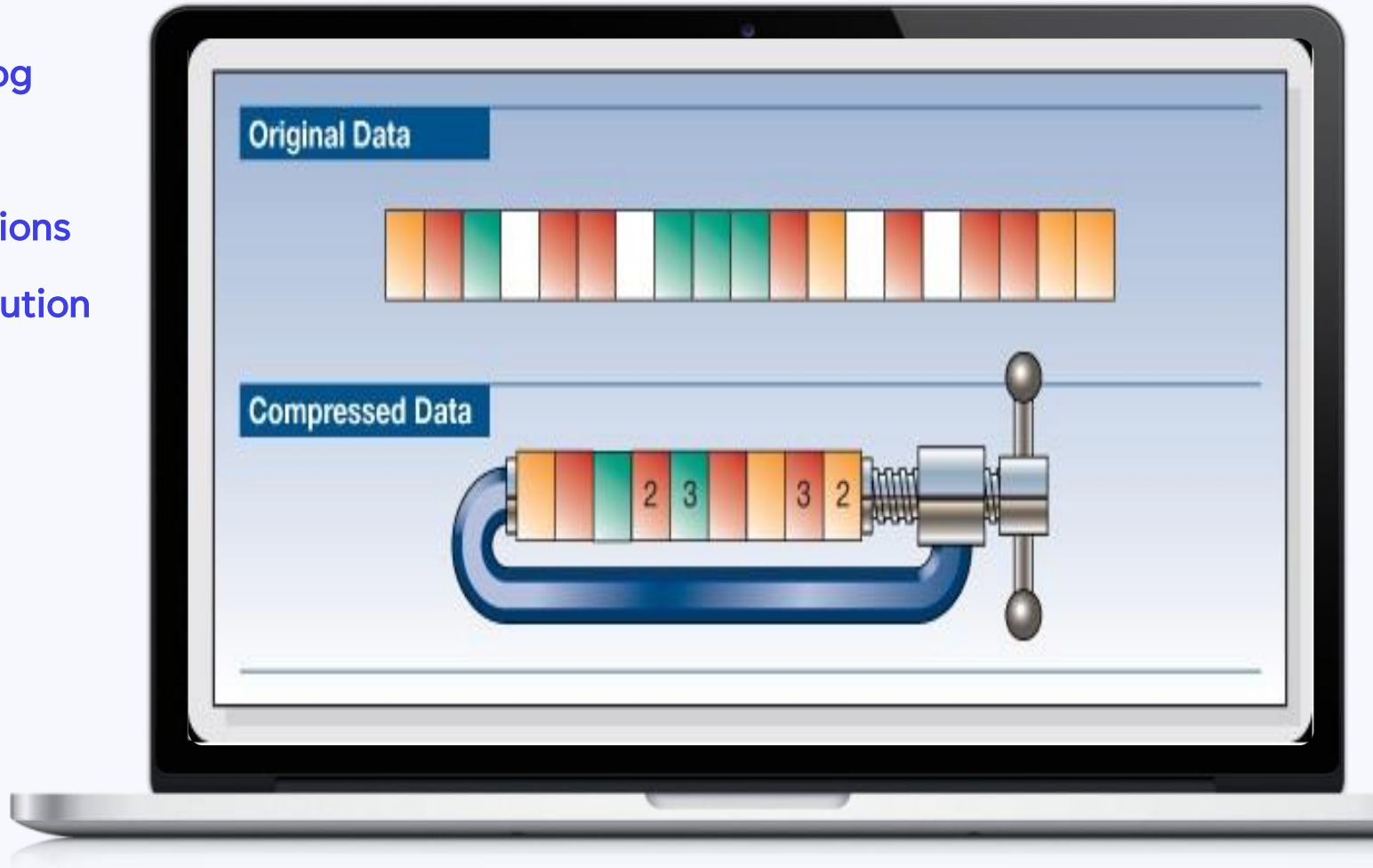
Why use compression?

→ Less DASD space, image copy and IMS Log volume used

→ Better buffer hit ratios for online transactions

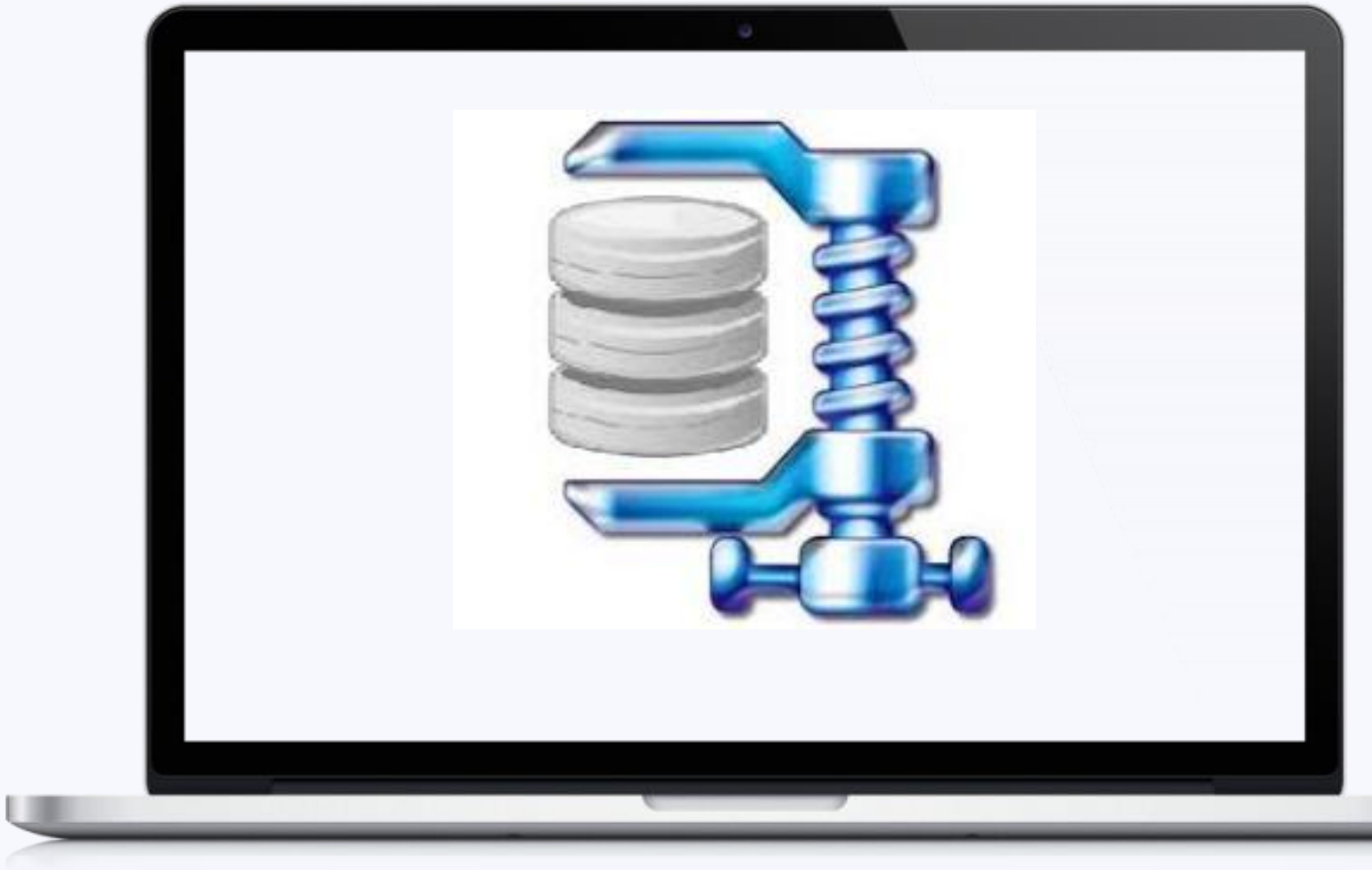
→ Improved performance during utility execution

→ NO application change



Data Packer/IMS Features

- → 8 Compression Techniques
- → IMS Space Management
  - → MSL – Minimum Segment Length
  - → FPS – Fixed Pad Size
  - → FSS – Free Space Segment Size
- → Trial Utility



→ Read More:

BMC AMI Data Packer for IMS™ Datasheet

DPK DBD implementation

→ **Full Function example**

- SEGM NAME=ROOT,BYTES=500, PARENT=0, COMPRTN=(DPIEXIT,DATA,INIT)
- DPIEXIT - is the compression routine used for Full Function Databases

→ **Fast Path example**

- SEGM NAME=ROOT2,BYTES=500,PARENT=0, COMPRTN=(DPIFPRTN,DATA,INIT)
- DPIFPRTN - is the compression routine used for Fast Path Databases.

→ **DATA -** is used to compress the data in the segment.

- KEY - can be used if you want to compress the key also (not recommended)
- IMS does not allow key compression for Fast Path databases.

→ **INIT -** IMS will call the compression routine when the database is opened.

Data Packer/IMS Benefits

→ **Extensive data integrity checks**

- Will NOT double compress

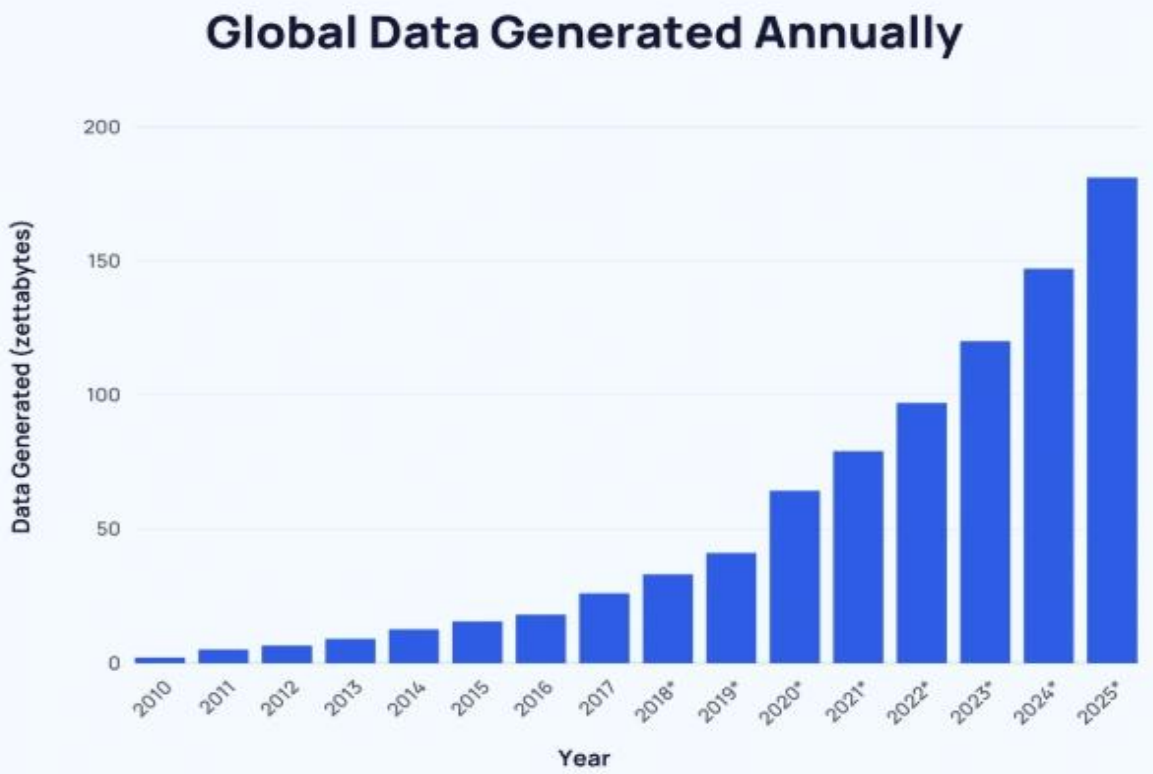
→ **Plain-text tolerance**

- Tolerates non-compressed data
- Segment inserted ASIS

→ **Easily administered**

- No registration required; can use all Product defaults
- 2 ways to for administrative tasks: ISPF and Batch JCL

Which way to go?



- Do nothing
- HALDB
- Fast Path
- PDF
- Compression

Depends on amount of time and resources available

Recent analyses indicate that approximately 90% of the world's data has been generated within the past two years, and according to IDC, the volume of data stored globally is doubling approximately every four years.
By 2025, the global volume of data is projected to rise further to 181 zettabytes by the end of 2025. This growth is driven by the increasing use of IoT devices, real-time data processing, and cloud-based storage.
Source: Article – Big data statistics: How much data is there in the world?

Reference materials

1. The Complete IMS HALDB Guide
    1. https://www.redbooks.ibm.com/redbooks/pdfs/sg246945.pdf
2. IMS Fast Path Solutions Guide
    1. https://docuri.com/download/fast-path-db_59c1d0c0f581710b28647b8c_pdf
3. BMC Partitioned Database Facility
    1. https://docs.bmc.com/docs/partitioneddatabasefacilityims/home-669653019.html
4. BMC Data Packer for IMS
    1. https://docs.bmc.com/docs/dpackerims/31/overview-783036919.html
5. IMS Primer
    1. https://www.redbooks.ibm.com/redbooks/pdfs/sg245352.pdf
6. Best Practices for HALDB – Peter Armstrong (2010)
    1. Contact gary_turner@bmc.com for a copy.

bmc

Thank you!