C13:

DB2 Deep Compression Experiences with SAP

Stefan Hummel
IBM Germany

IDUG
The Worldwide DB2 User Community

Wednesday, 07 October 2009
DB2 for LUW

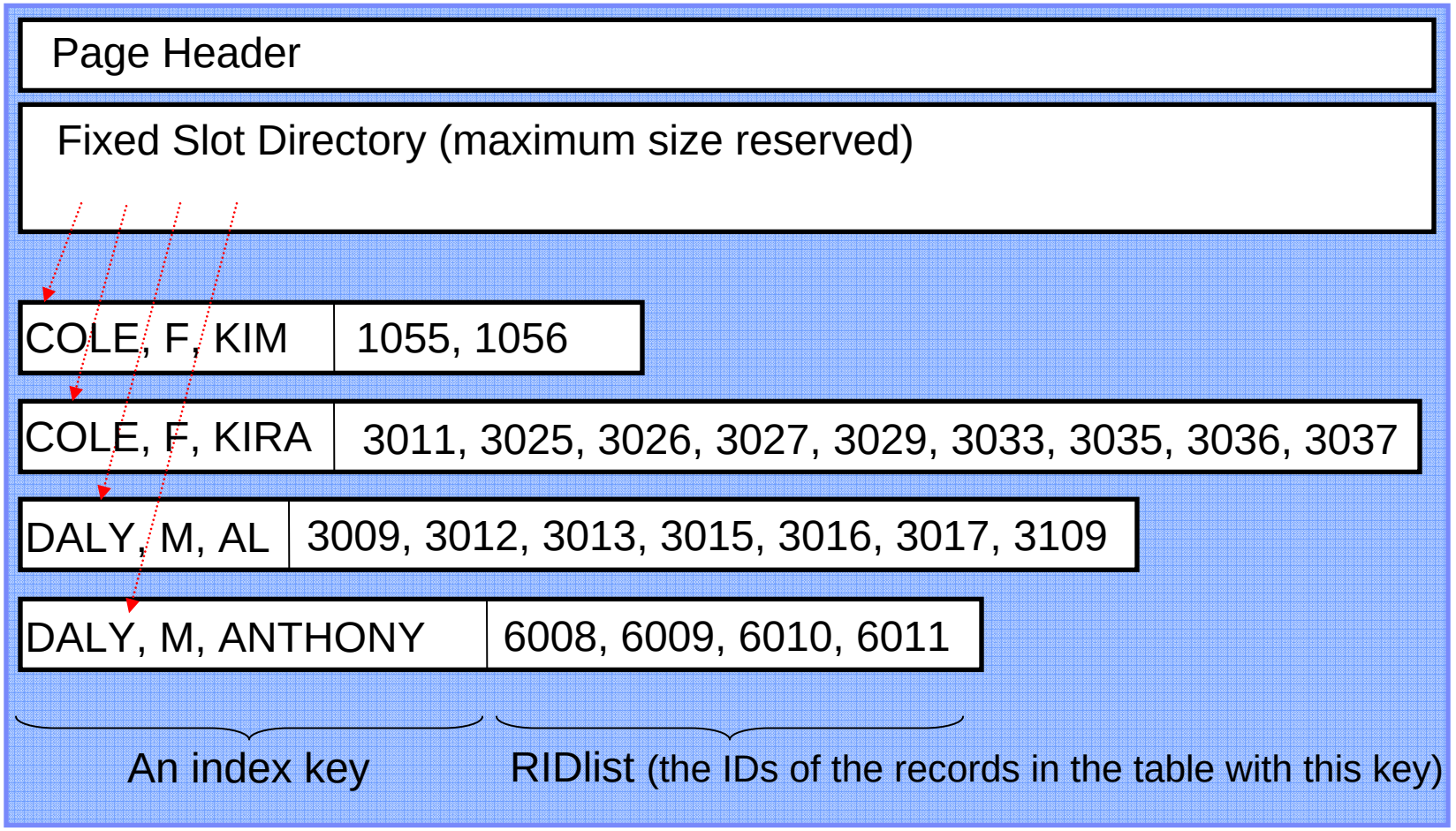
Agenda

- Index Compression
  - Algorythm
  - Estimation
  - Activation
  - Monitoring
- SAP heterogenous system copy including compression
- Customer Experiences
  - Installation and Activation
  - Storage Savings
  - Query Performance
- Discussion

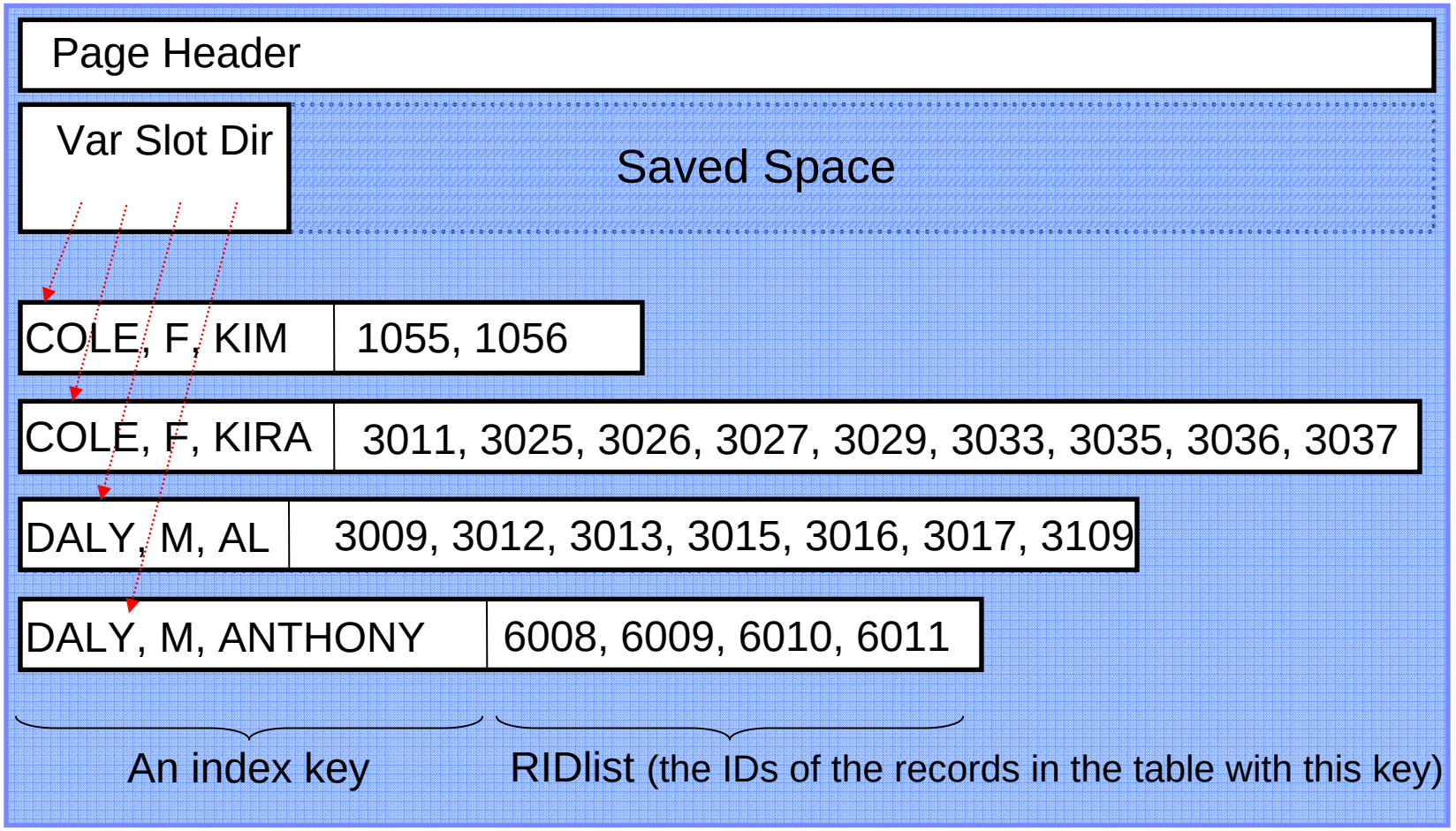
Index Compression

- Algorithms implemented by the Database Engine
  - RID List Compression
  - Prefix Compression
  - Variable Slot Directory
- Only data in leaf pages are compressed, not in the root page and any non-leaf pages in between
- DB2 uses a fixed algorithm, so no dictionary is required
- Applies to all indexes except:
  - Catalog indexes
  - MDC block indexes
  - XML path indexes and meta indexes
  - Index specifications

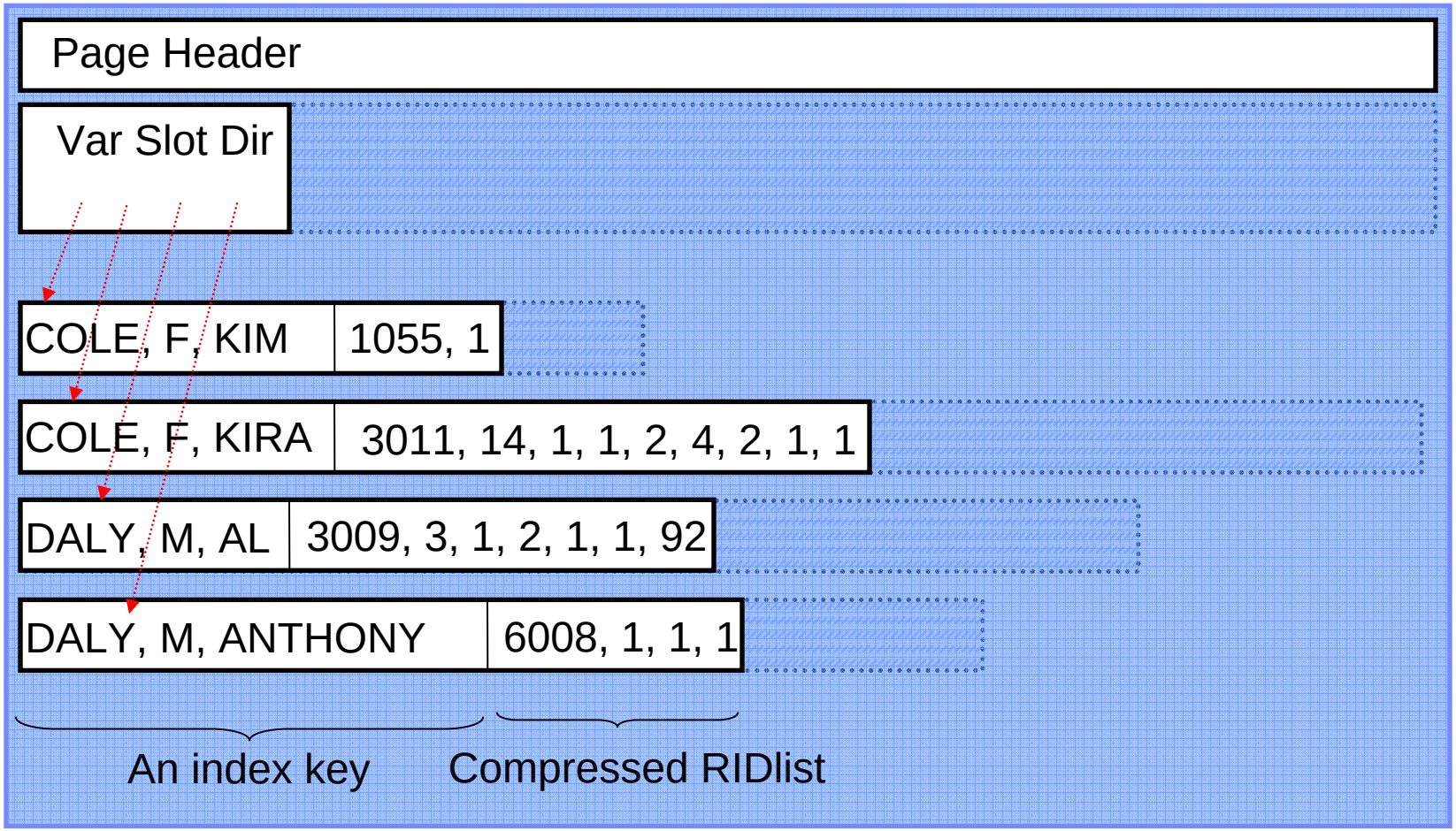
Existing Index Leaf Page Format



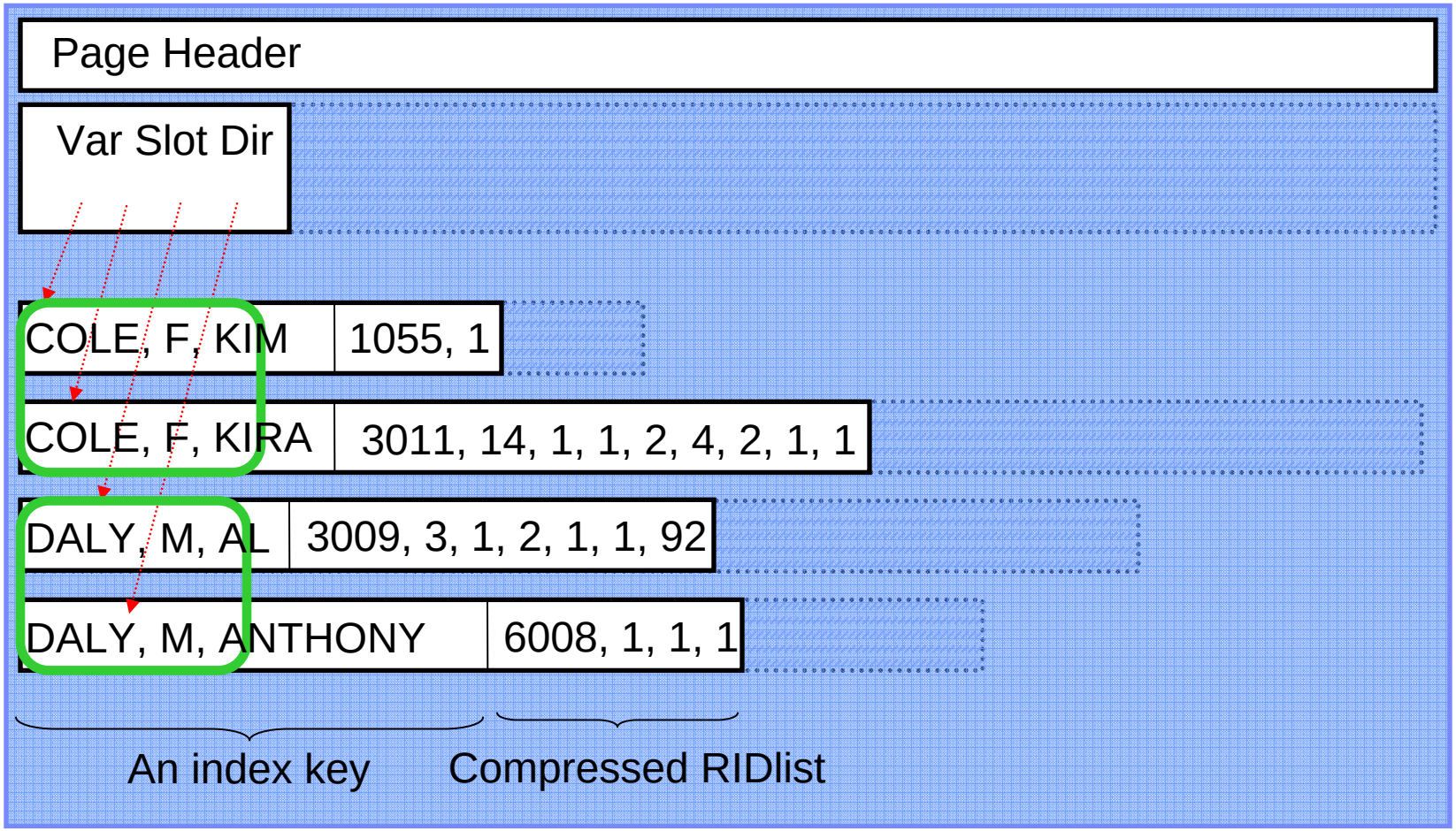
Variable Slot Directory



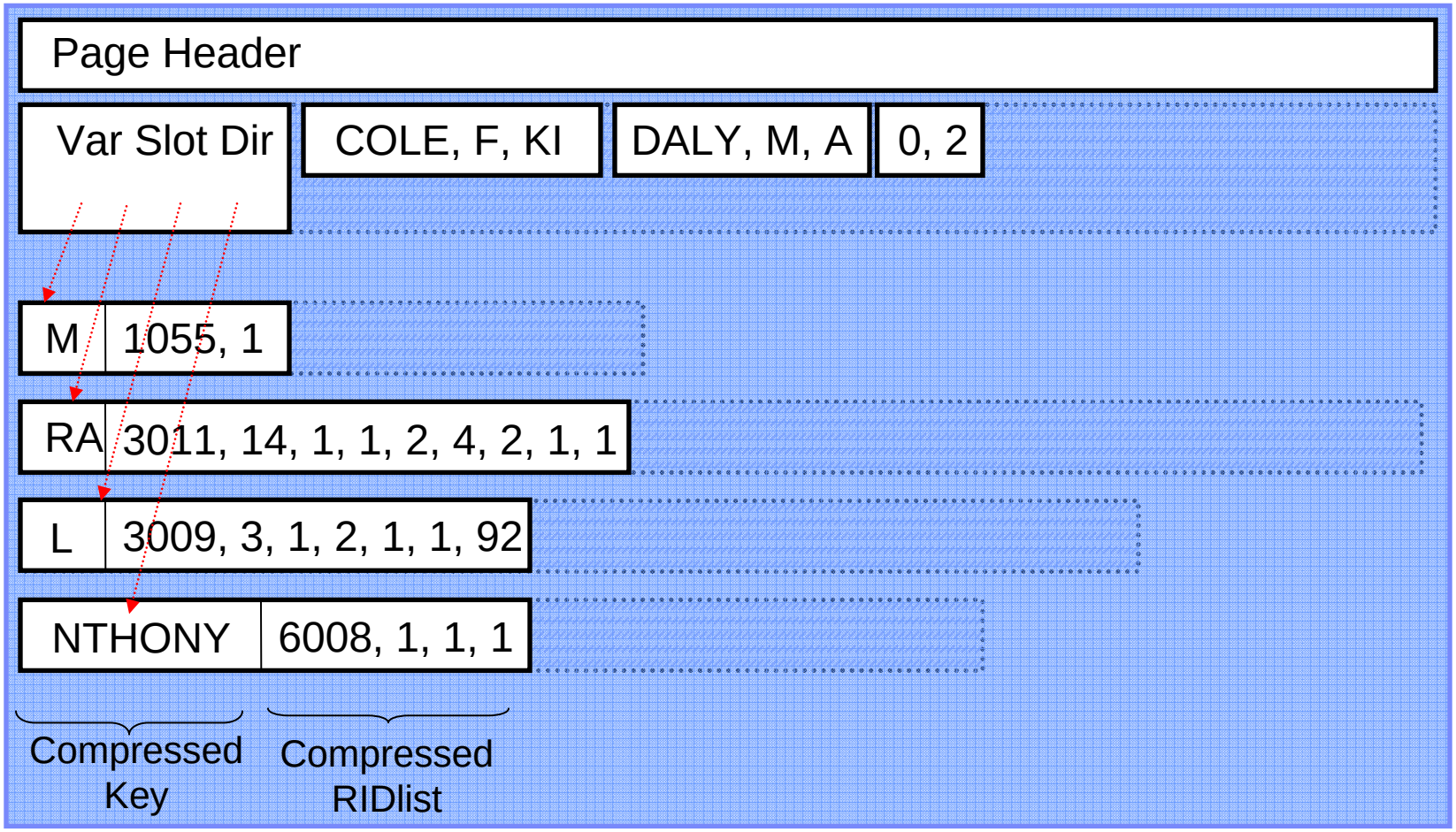
RIDlist Compression



Prefix Compression



Prefix Compression



Analyze database in terms of compression

- DB2 Compression Estimator
  Free tool to calculate the compression ratios for data ind indexes based on a non compressed database
- DB2 Table Function
  `ADMIN_GET_INDEX_COMPRESS_INFO` table function returns the potential index compression savings for uncompressed indexes or reports the index compression statistics from the catalog tables

```
TABNAME             INDNAME             PCT_PAGES_SAVED NUM_LEAF_PAGES_SAVED
------------------- ------------------- --------------- --------------------
EMPLOYEE            PK_EMPLOYEE                      47                  378
```

Index Compression Activation

- When row compression is activated on a table except for:
  - MDC block indexes, catalog indexes, index specifications
  - XML meta indexes, XML path indexes
- `CREATE INDEX` with the new "`COMPRESS YES`" option
- Via the new "`ALTER INDEX COMPRESS [YES|NO]`" statement, followed by an index reorg

Compression Savings Information

- ADMIN_GET_INDEX_COMPRESS_INFO table function returns the potential index compression savings for uncompressed indexes or reports the index compression statistics from the catalog tables

```
TABNAME             INDNAME             PCT_PAGES_SAVED NUM_LEAF_PAGES_SAVED
------------------- ------------------- --------------- --------------------
EMPLOYEE            PK_EMPLOYEE                      47                  378
```

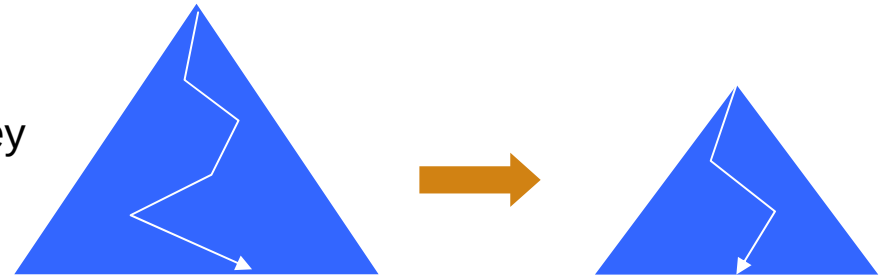
- ADMIN_GET_INDEX_INFO table function returns index information not available in the catalog views, such as compression information and the logical and physical size of the index

```
TABNAME             INDNAME             INDEX_OBJECT_P_SIZE
------------------- ------------------- --------------------
EMPLOYEE            PK_EMPLOYEE                         512
```

- COMPRESS and PCTPAGESSAVED in the SYSINDEXES catalog table show if an index is defined as compressed and the percentage saved respectively

Performance Attributes

- Fewer index levels
  - Fewer logical and physical I/Os for key search (insert, delete, select)
  - Better bufferpool hit ratio



- Fewer index leaf pages
  - Fewer logical and physical I/Os for index scans
  - Fewer splits
  - Better bufferpool hit ratio
- Tradeoff
  - Some additional CPU cycles needed for compress / decompress
    - 0-10% in early measurements
  - Typically outweighed by reduction in I/O resulting in higher overall throughput

SAP heterogenous system copy including compression

- R3load Version 7.00 and later offers a new SAMPLED compression option:
  1. Load representative sample of data into a table, e.g. 10%
  2. Builds a compression dictionary
  3. Load the complete set of data.
- Import into a compressed table is usually faster than into an uncompressed table (less write I/O)
- Phase 1 (sampling) can be done with data obtained from a test migration to save migration downtime

Customer Experiences

DB2 Compression Preparation

- Define tables and indexes to be compressed
  - Best practice: tables with NPAGES > 5.000 (16KB page size)
- Estimate compression ratio with `INSPECT` utility
- Formating output with `DB2INSPF`
- Generating an `ALTER` and `REORG` scripts (offline) for tables with estimated compression ratio > 40%
- Generating scripts for online `RUNSTATS` statements

Compression Savings (1)

- Clone of 500 GB SAP BI production system
- Compression of largest 36 tables
- Compression of all indexes
- 60% data storage saving
  - Tables size before compression: 264.902 MB
  - Tables size after compression: 106.916 MB
  - 157.986 MB savings
- 73% index storage savings
  - Indexes size before compression: 161.010 MB
  - Indexes size after compression: 42.723 MB
  - 118.287 MB savings
- 65% storage savings over all

Compression Savings (2)

- Clone of 500 GB SAP BI production system with reorganized indexes before compression
- Tablespace statistics before ix compression

WHAT	DATE	SIZE_IN_MB	FREE_SPACE_IN_MB
Indices	2009-05-12	214621	**69877**

- Tablespace statistics after ix compression

WHAT	DATE	SIZE_IN_MB	FREE_SPACE_IN_MB
Indices	2009-05-12	214462	**163543**

➔ ix compression ratio of ~ 65%

SQL Query Performance (1)

- SAP BI with compressed table data and indexes
  see page *“Compression Savings (1)”*
- Run of 12 SELECT statements taken from DB2 dynamic statement cache (DSC) of production system
  - statements with high executen time
  - statements without parameter markers
  - statements with result set bigger than 0 rows
- Hardware
  - IBM System p, P570
  - 2 x Power 5+ CPU’s
  - 8GB memory

SQL Query Performance (2)

- running the 12 statements like
  `SET SCHEMA ...`
  `SELECT ...`
  `COMMIT`
  with `db2batch` utility, each 3 times
- statements in red changed access path with compressed indexes

Total Time (s) ix compression	Run 1		Run 3	
	no	yes	no	yes
1	0.000	0.000	0.000	0.000
2	14.438	14.622	13.809	13.605
3	64.000	45.495	7.428	7.395
4	17.791	18.369	16.715	16.436
5	24.412	18.965	17.192	16.832
6	11.590	11.465	10.808	11.118
7	10.033	9.843	10.112	9.912
8	33.080	33.771	33.142	32.890
9	22.930	20.377	21.567	20.067
10	3.230	4.216	3.201	3.257
11	3.210	3.259	3.220	3.271
12	4.983	5.034	4.988	5.028
13	0.000	0.000	0.000	0.000

SQL Query Performance (3)

- Improvement of 11% with INSERT, UPDATE and DELETE statements comparing with no index compression
- Less CPU usage (compared to DB2 9.5) due to changes in the search algorithms for the primer and delta values

Type of Query	DB2 9.5 no ixcomp	DB2 9.7 no ixcomp	DB2 9.7 with ixcomp	Ratio Comp:Uncomp	Ratio DB2 9.5 / 9.7
Update1	1,324.79	1,248.54	1,077.70	0.86	0.81
Update2	220.34	186.50	185.54	0.99	0.84
Update3	334.45	297.11	259.64	0.87	0.78
Update4	689.09	573.79	526.87	0.92	0.76
Delete1	842.96	703.63	701.76	1.00	0.83
Delete2	144.20	121.71	108.97	0.90	0.76
Delete3	213.98	180.72	146.19	0.81	0.68
Delete4	429.03	368.20	295.67	0.80	0.69
Average	**406.85**	**350.20**	**312.27**	**0.89**	**0.77**

SQL Query Performance (4)

- `CREATE INDEX` of 3 indexes based on a single table
- Minimal elapse time increase due to calculation of delta values

Create Index	no ixcomp	with ixcomp	Ratio Comp:Uncomp
ix1	40,457	40,707	1.01
ix2	37,484	39,236	1.05
ix3	37,970	39,090	1.03

Backup Performance

- Backup, 1 image to filesystem, default parameters

Table compr.	Index compr.	Backup compr.	Size	Run [hh:mm]
no	no	no	500 GB	01:55
no	no	yes	112 GB	02:25
yes	no	no	278 GB	01:05
yes	no	yes	94 GB	02:00
yes	**yes**	**no**	**170 GB**	**00:29**
yes	yes	yes	62 GB	01:15

To consider, the original database size was 500 GB

Storage optimization ratio – customer experiences

DB2 9.7 Early Customer	Database Size	Data Compression Ratio	Index Compression Ratio	Total Database Savings
World leading construction machinery manufacturer, USA	725 GB	72%	49%	68%
Global consumer and commercial product marketer, USA	1.4 TB	58%	49%	56%
Haier Group, China	-	-	52%	-
John Deere, China	-	-	58%	-
Energy Delivery company, USA	62 GB	-	52%	-
Insurance company, Germany	176 GB	-	50%	-
T-Systems, Germany	500 GB	60%	73%	65%
Medical technologies company, USA	3.6 TB	-	65%	-

Recomendations

- Use table and index compression to
  - improve I/O efficiency
  - save storage space
  - save power and cooling
- Use backup compression if keeping the backup image on the filesystem
- Activate index compression initially
- Keep statistics up to date / use real-time statistics
- Monitor your system, keep an eye on:
  - compression rates
  - CPU usage

IDUG
The Worldwide DB2 User Community

Questions ?

further information

This presentation will be available for you to download in pdf format from the event website approximately one week after the event.

There are the following resources available to you on the IBM internet site:

- DB2DEMO, a program to demonstrate the functionality of IBM DB2
  http://www.ibm.com/developerworks/data/library/demos/db2demo/

Recommended reading:

- IBM DB2 product web site
  http://www.ibm.com/software/data/db2/9/
- DB2 blogs
  http://it.toolbox.com/blogs/db2luw
  http://db2expressc.blogspot.com

C13:

DB2 Deep Compression Experiences with SAP

Stefan Hummel
IBM Germany

IDUG
The Worldwide DB2 User Community

Wednesday, 07 October 2009
DB2 for LUW