

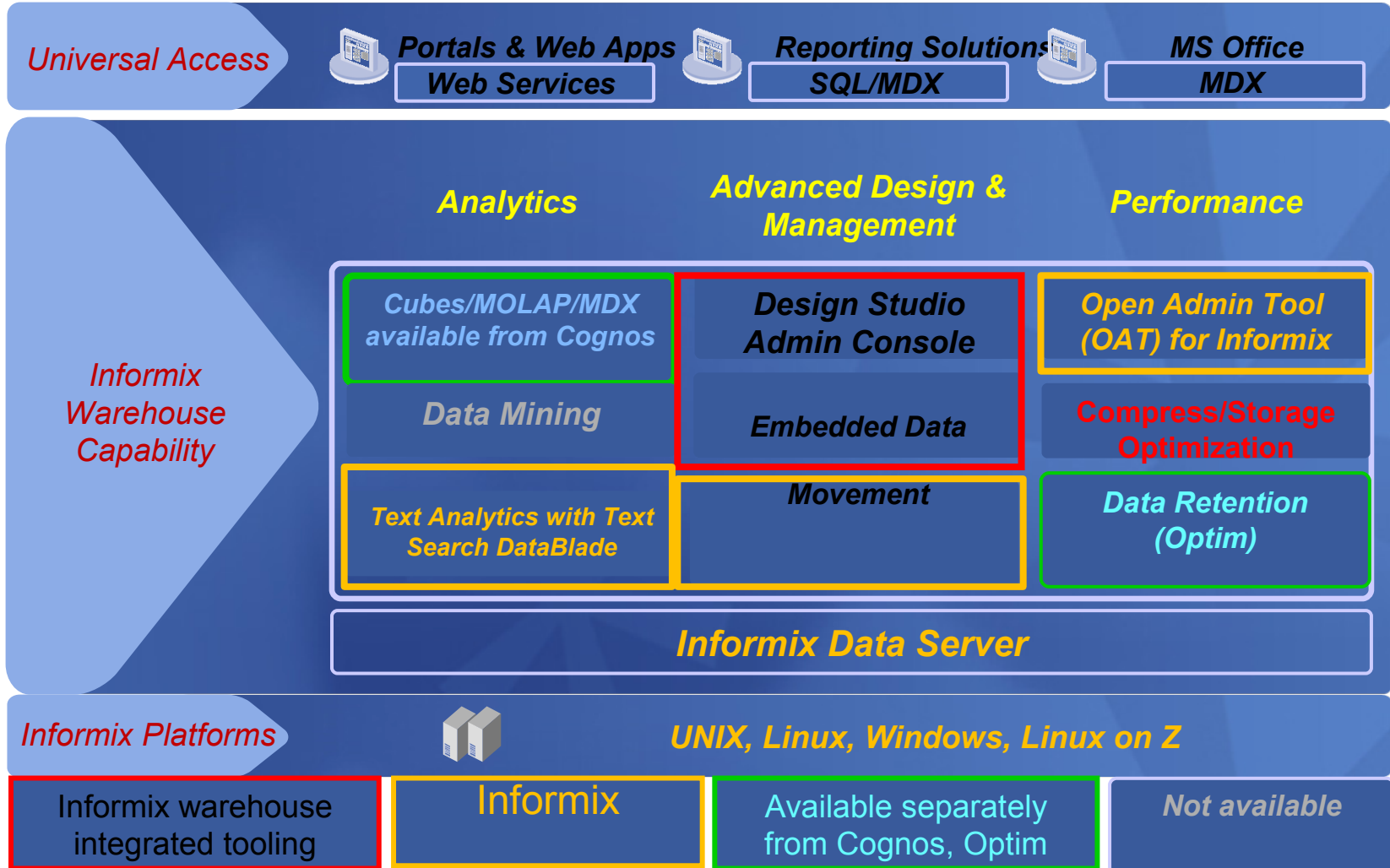


sandor.szabo@de.ibm.com

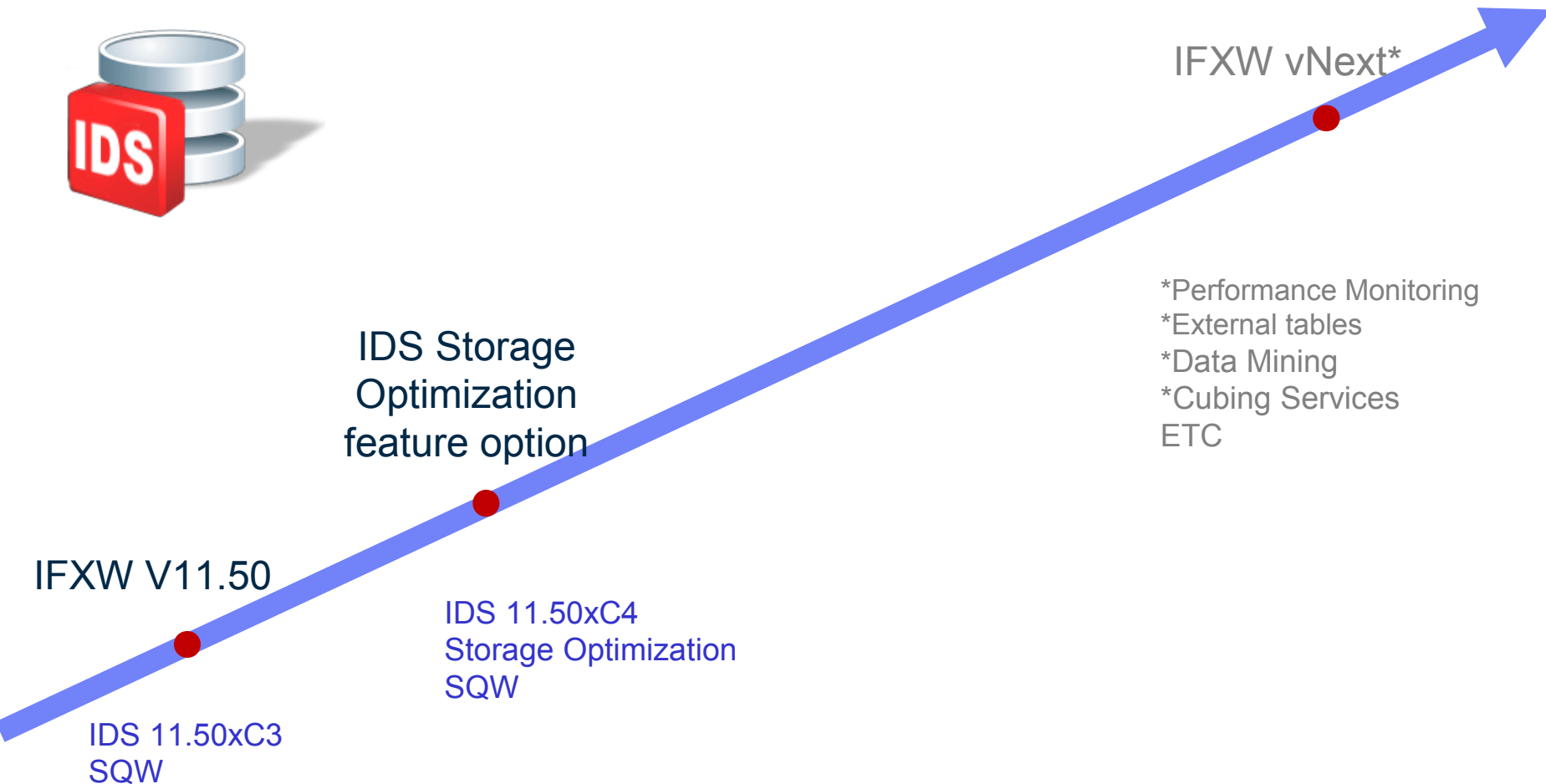
Informix Warehouse V11.50

IDS V11.50 + SQW

Announced on 3/5/2009



Informix Warehouse (IFXW) Roadmap



What Is Compression in IDS?

Store data rows in compressed format on disk

Saves up to 90% of row storage space

Ability to estimate possible compression ratio

Fits more data onto a page

Fits more data into buffer pool

Reduces logical log usage

Compress/uncompress data in a table or fragments

Time savings for backup and restore

Done with SQL administration API admin()

What Is Storage Optimization?

You can consolidate repack free space in a table or fragment and you can return this free space to the dbspace.

Space returned can then be used by any table in the dbspace

Compression Concepts

Lempel-Ziv (LZ) based algorithm – static dictionary, built by random sampling

Frequently repeating patterns replaced with 12-bit symbol numbers

Patterns can be up to 15 bytes long

Max possible compression = 90% (15 bytes replaced with 1.5 bytes = 12 bits)

Data Affects on Compression

Data with frequently repeating long patterns is the most compressible

Long runs of 0's or blanks are very compressible

Noise-like data is poorly or not at all compressible:

Encrypted data

Data already compressed by another algorithm

Data without long repeating patterns

Performance Impact of Compression

IO-bound workloads

Compression may improve performance by reducing IOs (both data page and logical log)

More data fits on a page, so more in buffer pool

Log records are smaller, so less logging

For CPU-bound workloads

Additional CPU used to compress and expand rows

Should not be a large impact

HDR, ER and Compression

All are supported on compressed tables

HDR

Tables will be compressed on secondary if they are compressed on primary

ER

Compression status of tables is independent between source and target, specified by user

Things That Cannot Be Compressed

Out-of-row data (e.g. blobs)

Indexes

Catalog tables

Temp tables

Partition tables

Dictionary tables

Tables in the following databases:

sysuser,sysmaster,sysutils,syscdr,syscdcv1

Dictionary Storage

Each compressed (non-fragmented) table or table fragment has its own compression dictionary

Dictionary consumes ~75K – 100K per fragment

Thus compressing tiny tables is not recommended

Admin API Interface

All compression and storage optimization operations are invoked via the IDS Admin API built-in UDRs

execute function task(...);

execute function admin(...);

Enables OAT graphical interface

Enables remote execution (DBA does not need to log directly in to the target machine)

Admin API Interface

Example

```
execute function task("table compress repack shrink",  
    "table_name", "database_name", "owner_name");  
execute function task("table repack shrink", "table_name",  
    "database_name", "owner_name");
```

```
EXECUTE FUNCTION task ("table  
    uncompress","tablename","dbname","owner name");
```

```
EXECUTE FUNCTION task ("fragment compress","list of  
    space separated partnums");
```

The database and owner names are optional.

Admin API Interface

Enable Compression SQL Administration

Required before first compress, uncompress, or
uncompress_offline

For estimate compression ratios, consolidate free space
(repack), return free space not needed.....

.

```
EXECUTE FUNCTION ADMIN ( " enable compression " );
```

Admin API Interface

estimate_compression SQL Administration

Estimates new compression ratio and a current ratio.

EXECUTE FUNCTION

```
task("estimate_compression","auto","insurance");
```

Admin API Interface

Create_dictionary **SQL Administration**

Creates a compression dictionary before compression

```
EXECUTE FUNCTION task("table create_dictionary", "rock",  
    "music", "mario");
```

If you do not create the compression dictionary as a separate step, IDS creates the dictionary automatically when you compress data.

Admin API Interface

Compress SQL Administration

Creates a compression dictionary (**implicit create_dictionary**)

Compresses all existing rows in-place

Space used to hold newly inserted rows without growing the table any larger.....

Table fully accessible to other queries

```
EXECUTE FUNCTION task("table compress","auto","insurance");
```

```
EXECUTE FUNCTION admin("fragment compress","14680071");
```

Admin API Interface

Repack **SQL Administration**

Table or fragment consolidates free space by moving data to the front

Use only with Isolation “Repeatable Read” or use **repack_offline**

```
EXECUTE FUNCTION task("table repack" ,"auto");
```

Admin API Interface

Repack_offline SQL Administration

Table or fragment consolidates free space by moving data to the front

table is XLOCKed, no query access

```
EXECUTE FUNCTION task("table repack_offline" ,"auto");
```

Admin API Interface

Shrink SQL Administration

Returns free space at the end of a fragment or table to the dbspace, thus reducing the total size of the fragment or table.

This operation is usually performed after a repack operation.

```
EXECUTE FUNCTION task("table shrink" ,"auto");
```

Admin API Interface

Uncompress **SQL Administration**

deactivates compression for new insert / update

uncompresses all compressed rows,

deactivates the compression dictionary.

allocates new pages for a fragment and moves

uncompressed rows that no longer fit

Use only with Isolation “Repeatable Read” or use

uncompress _offline

EXECUTE FUNCTION task("table uncompress" ,"auto");

Admin API Interface

Uncompress_offline SQL Administration

deactivates compression for new insert / update

uncompresses all compressed rows,

deactivates the compression dictionary.

allocates new pages for a fragment and moves

uncompressed rows that no longer fit

table is XLOCKed, no query access

```
EXECUTE FUNCTION task("table  
uncompress_offline", "auto");
```

Admin API Interface

purge_dictionary SQL Administration

No longer needed table/fragment dictionary deleted

Do this after you uncompress

Separate command because ER might need old dictionaries

```
EXECUTE FUNCTION task("table purge_dictionary" ,"auto");
```

```
EXECUTE FUNCTION admin("compression purge_dictionary",  
    "03/08/09");
```

Onstat -g dsk (print progress...)

Partnum	OP	Processed	Cur Page	Duration	Table
0x02900002	4	2793215	246952	158s	stock

OP One of the following flags that identifies the compression operation:

v 1 = create_dictionary

v 2 = compress

v 4 = repack

v 8 = repack_offline

v 16 = shrink

v 32 = uncompress

v 64 = uncompress_offline

v 128 = estimate_compression

v 256 = purge_dictionary

Processed: Number of rows processed so far for the specified operation

Curr Page: The current page number that the server is operating on now

Duration: The number of seconds that have elapsed since the operation started

Some “real” results from a Beta customer

TEST SUMMARY

Real Compression Rate to uncompressed Table: 79%

Online Compress/Repack/Shrink 7 Mio Rows: 20 Minutes

uncompress 7 Mio Rows: 10 Minuten

Some “real” results from a Beta customer

**INSERT into compressed Table is about 4% faster,
about 11% lesser CPU Resources ...**

**UPDATE compressed Table is about 32% faster,
about 6% lesser CPU Resources ...**

**UNLOAD compressed Table is as fast as
uncompressed Table – same CPU Resources ...**

**LOAD compressed Table is about 20% faster,
about 20% lesser CPU Resources ...**

OAT Interface

Compression - Mozilla Firefox: IBM Edition

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://klee:8080/openadmin/index.php?act=compression

Meistbesuchte Seiten The Informix Zone - W... heise online News Guy Bowerman informix - Google News IM Development BOE IBM IDSDevelopment SWG Technical Executi...

OpenAdmin Tool for IDS Server: utf8

Databases DBSpaces

tpcb

sysadmin

Table Name filter:

3 Tables for database: tpcb

Owner	Table	Page Size	Used Pages	Total Pages	Rows	Compressed	Usage
sandor	account	2 KB	2579	2608	49000	×	
sandor	teller	2 KB	258	275	4900	×	
sandor	branch	2 KB	2450	2464	4900	×	

Server Info

ServerType: Primary
Version: 11.50.FC4B5
ServerTime: 09:52:53
BootTime: 03-30 17:43
UpTime: 16:09:09
Sessions: 5

Übertragen der Daten von klee...

OAT Interface

Compression - Mozilla Firefox: IBM Edition

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://klee:8080/openadmin/index.php?act=compression

Meistbesuchte Seiten The Informix Zone - W... heise online News Guy Bowerman informix - Google News IM Development BOE IBM IDSDDevelopment SWG Technical Executi...

OpenAdmin Tool for IDS Server: utf8

Home

- Health Center
- Alerts Dashboard
- Logs
 - Admin Command
 - Online Messages
 - OnBar Activity
- Task Scheduler
 - Scheduler
 - Task Details
 - Task Runtimes
- Space Administration
 - DBSpaces
 - Chunks
 - Recovery Logs
 - Compression
- Server Administration
 - MACH
 - Configuration
 - System Validation
 - User Privileges
 - Virtual Processors
 - Auto Update Statistics
- Enterprise Replication
- Performance Analysis
 - SQL Explorer
 - Performance History
 - System Reports
 - Session Explorer
- SQL ToolBox
 - Databases
 - Schema Browser
 - SQL Editor
- Help
 - Useful Links
 - How do I?
 - Readme
 - About OAT
 - ER Plug-in: How do I?
- Admin
- Logout

Server Info

ServerType: Primary
Version: 11.50.FC4B5
ServerTime: 09:52:53
BootTime: 03-30 17:43
UpTime: 16:09:09
Sessions: 5

Databases DBSpaces

tpcb Table Name filter:

Compress 3 Tables for database: tpcb

Owner	Table	Page Size	Used Pages	Total Pages	Rows	Compressed	Usage
sander	account	2 KB	2578	2608	48000	X	15
sander	teller	2 KB	208	272	4800	X	15
sander	branch	2 KB	2450	2464	4800	X	15

tpcb:sander.teller

☒ Compress Build a compression dictionary and compress the selected table.

☐ Repack Fill any holes by moving rows to the front of the fragment.

☐ Shrink Remove free extents.

OK Cancel

Übertragen der Daten von klee...

Typical Examples

1. **EXECUTE FUNCTION** task("enable_compression");
2. **EXECUTE FUNCTION** task("table estimate_compression",
"rock", "music", "mario");
3. **EXECUTE FUNCTION** task("table create_dictionary", "rock",
"music", "mario");
4. **EXECUTE FUNCTION** task("table compress repack shrink",
"rock", "music", "mario");
5. **EXECUTE FUNCTION** task("table uncompress", "rock",
"music", "mario");
6. **EXECUTE FUNCTION** task("table purge_dictionary", "rock",
"music", "mario");

Attention !

Compress, repack, repack_offline, uncompress, and uncompress_offline operations can consume large amounts of logs

The fragment or non-fragmented table must contain at least 2,000 rows to compress

You cannot perform compression operations on an HDR secondary, RSS secondary, or SDS secondary server

Summary

Significant savings in disk storage space €€€€€

I/O-bound tables, for example, those with bad cache hit ratios, are good candidates for compression.

Compression reduces logging

Compression fits more data into the buffer pool

Storage Optimization allows space saved by compression to be reclaimed from tables and fragments of tables

Questions

