Information Management

Section 4

Supporting Compression and Storage Optimization

Informix. software

Oncheck –pt/T changes

- Oncheck –pt/T shows compression dictionary information:

```
Compression Dictionary Identifiers
         rowid     loguniq      logpos
          2502          89      b6c04c
```

- Oncheck –pT shows a data summary for compressed tables:

```
Compressed Data Summary

       Number of  rows                      8192
       Number of compressed rows            8192
       Percentage of compressed rows         100.00
```

Dictionary table

- Each compressed (non-fragmented) table or fragment has its own compression dictionary
- Dictionary consumes approximately 75K – 100K per fragment
  - thus compressing tiny tables is not recommended
- All dictionaries for tables/fragments in a given dbspace are stored in a special hidden dictionary table in that dbspace
- The dictionaries for active compressed partitions are placed into memory and cached for later access. Why? Performance!

Dictionary table structure

- Table structure:

dict_partnum	integer,
dict_version	integer,
dict_dbsnum	integer,
dict_create_timestamp	integer,
dict_create_loguniqid	integer,
dict_create_logpos	integer,
dict_drop_timestamp	integer,
dict_drop_loguniqid	integer,
dict_drop_logpos	integer,
dict_dictionary	byte

- Lock mode row
- Two attached indices:
  - Key 1: Unique composite index on dict_partnum, dict_drop_loguniqid, dict_drop_logpos
  - Key 2: Duplicate single-column index on dict_partnum

Dictionary table – viewing through sysmaster

- A view of all dictionary tables for the instance
  - syscompdicts_full table – includes binary dictionary; access restricted to user “informix”
  - syscompdicts view – globally accessible; omits binary dictionary for security

Dictionary table – onstat -g ppd

- In-memory view of all active dictionaries on the instance

```
Partition Compression Dictionary Info
partnum    Version  DbsNum   CrTS     CrLogID  CrLogPos DrTS     DrLogID  DrLogPos
0x200002   1        2        1238367659 7        7356492  0        0        0
```

Removing compression – purge_dictionary

- Deletes old **inactive** dictionaries
- This is a separate command because ER or CDC (DataMirror) might need old dictionaries for log records not yet snooped/replayed

Removing compression – purge_dictionary (cont)

- Deletes old **inactive** dictionary entries for a specified table name:
- `execute function sysadmin:task|admin(“table purge_dictionary”,“database_name”,”table_name”,”owner_name”);`
- NOTE: this will fail if the table has been dropped.

- Deletes old **inactive** dictionary entries for a specified partnum or list of partnums:
- `execute function sysadmin:task|admin(“fragment purge_dictionary”,“partnum_list”);`

Removing compression – purge_dictionary (cont)

- Deletes all old **inactive** dictionaries:

-
  ```
  execute function sysadmin:task|admin(“compression
  purge_dictionary”);
  ```

- Deletes all old **inactive** dictionaries created prior to the specified date.

-
  ```
  execute function sysadmin:task|admin(“compression
  purge_dictionary”,”date”);
  ```

Table level restore of dictionary table

- Can the compression dictionary table itself be restored with TLR?
  - No, because that method requires direct SQL access to the table. The only SQL access is read-only through the syscompdicts_full SMI table which is a view to shared memory structures and is not really archived.
- The compression dictionary table is archived with the dbspace and can be restored at the dbspace or instance level.
- For the above reasons and those unveiled in later slides, customers should be doing periodic unloads of the sysmaster:syscompdicts_full table.

Viewing logical logs with compressed row data

- If a table/fragment is compressed, the row data in the logical logs will be compressed. By default, the log viewer tools will display the data in the logical logs as it exists (compressed).
- Onlog can display current or archived (with ontape) log records in both compressed and uncompressed form. To view uncompressed:
  - onlog <options> -l –c *compression_dictionary_file*
- Onbar can display archived (with onbar) log records in both compressed and uncompressed form. To view uncompressed:
  - onbar <options> -l –c *compression_dictionary_file*
- *compression_dictionary_file* is an unload file from the sysmaster:syscompdicts_full table

Section 5

Debugging Compression and Storage Optimization

Orphaned compression dictionary entry

- May prevent a drop of the dbspace.
- Caused by dropping a compressed table/fragment without first uncompressing the table/fragment.
- To avoid: uncompress the table before dropping the table
- To correct: use appropriate purge_dictionary command to remove the table/partition’s dictionary entry
- The compression dictionary table (and possibly rows in the table) will still exist even if the dbspace is empty. The code responsible for dropping a dbspace, should detect and handle this situation.

Compression dictionary partition dropped or page corrupt

- -242/-135 errors when accessing (select, insert, update, delete) or dropping the compressed table/fragment
  - It may be necessary to set onmode –I 135 to find the partnum that rscRetrieveLatestDict() is trying to open.
- Message log reports the following error message:
  - **`rscLoadLatestDict: dictionary retrieval failed`**
    - idsdb00174134 reported to request more info (fix?)
- DSD might be able to repair/salvage the compressed table/fragment
  - More likely than not if the customer has a recent unload of the syscompdicts_full table

Missing compression dictionary entry

- -242/-135 errors when accessing (select, insert, update, delete) or dropping the compressed table/fragment
  - It may be necessary to set onmode –I 135 to find the partnum that rscRetrieveLatestDict() is trying to open.
- Message log reports the following error message:
  - **`rscLoadLatestDict: dictionary retrieval failed`**
    - idsdb00174134 reported to request more info (fix?)
- DSD might be able to repair/salvage the compression dictionary row
  - Very likely if the customer has a recent unload of the syscompdicts_full table

Compression dictionary entry - data/index page header corruption

- -242/-135 errors when accessing (select, insert, update, delete) or dropping the compressed table/fragment
  - It may be necessary to set onmode –I 135 to find the partnum that rscRetrieveLatestDict() is trying to open.
- Message log reports the following error messages:
  - **`Assert Failed: Page Check Error in`** …
    - Action: Run ‘oncheck –cD/I dbspace_name:”informix”.rscompdict…
    - The above command will not work. Why?
    - Oncheck –cI run against the dictionary table partnum will likely not repair a bad index.
  - **`rscLoadLatestDict: dictionary retrieval failed`**
    - idsdb00174134 reported to request more info (fix?)
- DSD might be able to repair/salvage the compression dictionary row
  - Very likely if the customer has a recent unload of the syscompdicts_full table

Would you like to know more?

- http://www.ibm.com/software/data/informix/compression
- http://en.wikipedia.org/wiki/LZ77_and_LZ78

Defragment (from 11.70)

You can improve performance by defragmenting partitions to merge non-contiguous extents.

To determine how many extents a table, index, or partition has, you can run the oncheck -pt and -pT command.

To defragment a table, index, or partition, run the SQL administration API task() or admin() function with the **defragment** argument or the **defragment partnum** argument and specify the table name, index, or partition number that you want to defragment.

- EXECUTE FUNCTION task(“defragment”,“stores_demo:informix.customer");
- EXECUTE FUNCTION admin(“defragment partnum”, “16777217,28477346");

Defragment - Limitations and considerations

Before you defragment a partition, review these important considerations:

You cannot stop a defragment request after the request has been submitted.

You cannot be defragment the following objects:

– Pseudo tables, such as Virtual-Table Interface (VTI) tables

– Tables with Virtual-Index Interface (VII) indexes

– Tables with B-tree functional indexes

– Temporary tables

– Sort files

Defragment - Limitations and considerations

You must not issue conflicting operations on a table or partition that you want to defragment. The first operation must complete before the second operation is started. If the first operation is still running, the request for the second operation returns an error. The following list contains examples of conflicting operations:

– Only one defragment request can operate on a partition at a time.

– Only one defragment request can operate on a dbspace at a time.

– A table cannot be defragmented when DDL statements, such as DROP TABLE or ALTER FRAGMENT, are being run on the table or partition.

Defragment - Limitations and considerations

– A table cannot be defragmented when the table is being truncated.

– A table cannot be defragmented when the table is being compressed or uncompressed.

– A table cannot be defragmented when an online index build is being run.

– A table cannot be defragmented that has exclusive access set on the table.

If there are problems completing a defragment request, error messages are sent tothe online log file.

Example …..

Defragment - Limitations and considerations

Before you defragment a partition, review these important considerations:

You cannot stop a defragment request after the request has been submitted.

You cannot be defragment the following objects:

– Pseudo tables, such as Virtual-Table Interface (VTI) tables

– Tables with Virtual-Index Interface (VII) indexes

– Tables with B-tree functional indexes

– Temporary tables

– Sort files