Index Rebuild for Growth

Paul Mak

May, 2016

Contents

Here is a description of how to rebuild an index for growth to avoid causing index problem if not properly sized. We learned our lesson from a previous event.

Previous Problem

The index area PRD.FPES003X-AREA was expanded using unload/reload on March 14th 2016 which took 6 hours to rebuild the index SYS-DBMGVC-ID. All orphan records were removed. By 12.30pm March 15th the orphan records have gone from zero to 287488.

User program such as DCAU394 started to experience performance problem on the index. It would normally take 5 minutes to complete but is now taking 3 hours and had to be cancelled.

The underlying problem

SR8 size 8072 is greater than 7040 (30% of page size).
The solution is to either decrease the INDEX BLOCK CONTAINS value or increase the area page size.

We decided to increase the page size from 23476 to 27996. The XPAGE (Expand File) job took 2 minutes
Now the SR8 size of 8072 is smaller than 30% of 27996=8398

With the page size change

In the morning:	Processing 2000 transactions took 3 hours, 3000 Calls to IDMS, and 1 million Page Read
After the page size change:	Processing 6000 transactions took 6 minutes, 30,000 Calls to IDMS and 2000 Page Read

An orphaned indexed record is a record whose index pointer does not point back to the index record (SR8) that contains the record's index entry. Orphans occur as the result of splitting an existing SR8 into two records to accommodate a new entry. When an SR8 contains the maximum number of entries and a new member targets to that SR8, it must split.

As part of the split, some of the entries are moved to a new SR8, but the index pointer in their associated records are not adjusted to reflect the change, resulting in “orphaned” records.

The high "orphan counts" which are the result of SR8 splits in rapidly growing system-owned linked indexes cause increasing performance problem during erase, disconnect, and purge.

The database is updated with a large number of new index members on a daily basis by DCAU394. It typically add, erase or disconnect members.

The "Rebuilding for growth" technique below will delay splitting many SR8s in the index thus reducing the performance problem when disconnecting or erasing members from the index.

Rebuilding for Growth

Rebuilding for Growth uses a combination of Page Reserve and a smaller Block Contains value than the one used for normal processing. This results in SR8s that are able to handle a certain number of new members without splitting.

The Page Reserve is space that can be used for the SR8s to expand but will prevent the storing of new SR8s.

When the MAINTAIN INDEX utility rebuilds an index it will create SR8 records that contain the maximum number of entries as specified by the Index Block Count (IBC). As a result, the first time a new record is added to the index its target SR8 must be split.

Since the page size of most index areas is selected to hold a specific number of full-size SR8 records the new SR8 created during the split will also be overflowed to some other page within the area. This means that the first insertion has created a situation that will impact on the performance of that index.

To avoid index's performance problem is to avoid splitting the SR8 records. The index is rebuilt using an IBC that is smaller than the desired maximum and assign a Page Reserve to the index area. For example, assume the required regular IBC is 80 and the page size is 11476 for the index area that will hold 3 SR8 records. First define the IBC is set to 70.

The difference 400 bytes between the required space for an IBC of 80 (SR8 Size 3199) and an IBC of 70 (SR8 size 2809) should be assigned as the Page Reserve for the area in a DMCL.
Since the runtime environment specifies the IBC to be 80, each SR8 record can have 10 new records inserted before the need to split the SR8 arises. The runtime environment would not use the Page Reserve so as the SR8 records grow there will be enough space per page for each SR8 to grow to its maximum IBC.

From the print index report:
SR8 size 3199 is less than 3442 (30% of page size 11476).
The INDEX BLOCK CONTAINS 80 is good

What happens when SR8 Greater than 30% of Page Size

This is what happens if SR8 size is greater than 30% of page size .

Each SMP has a two-byte Space Management Entry for each page it controls. That two byte-entry contains (11476 - 32) until the page in question encounters a STORE Failure and is over 70% full. At this time the Space Management Entry is changed from (11476 - 32) to the actual Space available on the page.

When a record being STORE cannot fit on its target page we read the corresponding SMP and look for the Next page in the area that indicates it has sufficient space for the record being stored.

If the index area violates the recommendation about record lengths such that pages get to a point where we cannot fit any more record occurrences on the page, BUT the page is NOT over 70% full.

Each page fits a number of SR8 and leave some bytes free. If it is not enough to hold another SR8, problem will arise if the page can never get over 70% full.

For such pages the corresponding Space Management Entry on an SMP will never reflect the actual Space available on that page. It will continue to contain (11476 - 32) and appear as if there is space available for storing new records.

What happens when an SR8 Will Not Fit the Target Page

We have a new SR8 to be STORE but will not fit on its target page.

We read the SMP and see that the next page in the area has enough space available to fit the new SR8 (because it shows 11476-32).

We read that page, BUT find that it actually cannot fit the record.

We go back to the SMP and see that the subsequent page can fit the record...
BUT that won't fit it either...

This can go on and on if you get to a point where many pages cannot fit the new record but are not over 70% full.

Are we surprise to find over 1 million page read and counting?

ALWAYS be certain all pages in an area CAN get to be over 70% full when storing SR8 records to avoid the "SMP I/O thrashing"

Print Index After Expansion – Best Practice

It is good practice to run IDMSBCF print index following an expansion of an index area using Unload/Reload. This job must be run in Local Mode for performance reason

```
//SYSIPT   DD *
PRINT INDEX SYS-PEPN-ID SEGMENT PED USING SS0004 SUMMARY;
/*
```

Check the end report for any Performance Hint such as below, for example:

Performance hint: SR8 size 8072 is greater than 7040 (30% of page size).
Decrease the INDEX BLOCK CONTAINS value or increase the area page size.

Page Reserve

Page reserve is the amount of space on a page that is used only for the expansion of existing records or rows. It is never used for storing new occurrences.

Specifying a page reserve as part of an area definition is useful if the area contains:
- Indexes
- Variable length records
- Compressed records or rows

The page reserve for an area can also be specified as an area override within a DMCL definition. Specifying it at the DMCL level allows tailoring the page reserve for particular types of processing, such as database loading or index building. By specifying a page reserve during these types of operations and then reducing or removing it altogether, you ensure that each page will contain free space for both new record occurrences or rows and the expansion of variable length objects.

DMCL Definition for Rebuild

This is the DMCL definition used to rebuild SYS-PEPN-ID index in PED (pre-production):

```
CREATE
  PHYSICAL AREA PED.FFER011X-AREA
          PRIMARY SPACE 1000 PAGES  FROM PAGE 2150001
```

```
            MAXIMUM SPACE 1000 PAGES
            PAGE SIZE 11476 CHARACTERS
            PAGE RESERVE 400 CHARACTERS
        INCLUDE SYMBOLIC INDEX SYS-PEPN-ID
            BLOCK CONTAINS 70 KEYS
        WITHIN FILE FFER011X
            FROM 1 FOR ALL BLOCKS
        ;


//SYSIDMS  DD *
ECHO=ON
DMCL=IDMSDMCN
LOCAL_DYNAMIC_ALLOCATION=OFF
//SYSIPT   DD *
  MAINTAIN INDEX IN SEGMENT PED USING SS0004
  REBUILD SYS-PEPN-ID   FROM ALLROWS ;
```

Print Index after Rebuild

```
Print Index Report after the index rebuild

  CONNECT TO PEDDICT;
 Status = 0        SQLSTATE = 00000

  PRINT INDEX "SYS-PEPN-ID" SEGMENT PED
      USING SS0004
        SUMMARY ;
 Status = 0        SQLSTATE = 00000
-
SET Name: SYS-PEPN-ID
    IBC 70                              Displacement        0
    Sort option SORTED SYM ASC          Key length          31
    Duplicates LAST                     Compression         No

 OWNER: SR7
    AREA PED.FFER011X-AREA              Low page            2150001
    Page size 11476                     High page           2151000

 MEMBER: PEPN-TRIGGER                   Set membership      Mandatory
Automatic
    Located VIA index No                Index is            Linked
    AREA PED.FFER002D-AREA              Low page            1806001
    Page size 6356                      High page           1808497

 OWNER X'20CF3D01' on page 2150205
    Top level SR8  on page 2150228      utilization      7.1%

 Intermediate Level
    Nr of SR8s                                 6                6 Minimum
    Nr of pages with SR8s                      6                2 Minimum
    Nr of displaced SR8s                       0       0.0%
    Nr of entries in use                     346      82.3%
    Nr of Orphans                              0       0.0%
    Total size of all SR8s                 16824

 Bottom       Level
```

```
    Nr of SR8s                                341                341 Minimum
    Nr of pages with SR8s                     115                114 Minimum
    Nr of displaced SR8s                        0       0.0%
    Nr of entries in use                    23870     100.0%
    Nr of Orphans                               0       0.0%
    Total size of all SR8s                 956104

 Index occurrence totals
    Nr of members                           23870
    Nr of levels                                3                  3 Minimum
    Size of largest SR8                      2804
    Nr of SR8s                                347                347 Minimum
    Nr of pages with SR8s                     115                116 Minimum
    Nr of displaced SR8s                        0       0.0%
    Nr of entries in use                    24216      99.6%
    Nr of Orphans                               0       0.0%
    Total size of all SR8s                 972928

 Nr of Buffers versus Estimated IOs for Sequential Bottom Level access
 -------------        -------------
      1  - 20                  115
```

DMCL Definition for Growth

This is the DMCL definition used to maintain SYS-PEPN-ID index for growth in PED:

Notice that the Block Contains 70 keys is now changed to 80 keys and the Page Reserved is removed

```
    ALTER
    PHYSICAL AREA PED.FFER011X-AREA
            PRIMARY SPACE 1000 PAGES  FROM PAGE 2150001
            MAXIMUM SPACE 1000 PAGES
            PAGE SIZE 11476 CHARACTERS
        INCLUDE SYMBOLIC INDEX SYS-PEPN-ID
            BLOCK CONTAINS 80 KEYS
        WITHIN FILE FFER011X
            FROM 1 FOR ALL BLOCKS
        ;
```

Print Index

```
  CONNECT TO PEDDICT;
 Status = 0        SQLSTATE = 00000

PRINT INDEX "SYS-PEPN-ID" SEGMENT PED
      USING SS0004
        SUMMARY ;
 Status = 0        SQLSTATE = 00000
-
SET Name: SYS-PEPN-ID
    IBC 80                              Displacement        0
    Sort option SORTED SYM ASC          Key length          31
    Duplicates LAST                     Compression         No

 OWNER: SR7
```

```
    AREA PED.FFER011X-AREA              Low page            2150001
    Page size 11476                     High page           2151000

 MEMBER: PEPN-TRIGGER                   Set membership      Mandatory
Automatic
    Located VIA index No                Index is            Linked
    AREA PED.FFER002D-AREA              Low page            1806001
    Page size 6356                      High page           1808497

 OWNER X'20CF3D01' on page 2150205
    Top level SR8  on page 2150228      utilization      6.2%

 Intermediate Level
    Nr of SR8s                                 6                5 Minimum
    Nr of pages with SR8s                      6                2 Minimum
    Nr of displaced SR8s                       0       0.0%
    Nr of entries in use                     346      72.0%
    Nr of Orphans                              0       0.0%
    Total size of all SR8s                 16824

 Bottom       Level
    Nr of SR8s                               341              299 Minimum
    Nr of pages with SR8s                    115              100 Minimum
    Nr of displaced SR8s                       0       0.0%
    Nr of entries in use                   23870      87.5%
    Nr of Orphans                              0       0.0%
    Total size of all SR8s                956104

 Index occurrence totals
    Nr of members                          23870
    Nr of levels                               3                3 Minimum
    Size of largest SR8                     2804
    Nr of SR8s                               347              304 Minimum
    Nr of pages with SR8s                    115              101 Minimum
    Nr of displaced SR8s                       0       0.0%
    Nr of entries in use                   24216      87.2%
    Nr of Orphans                              0       0.0%
    Total size of all SR8s                972928

 Nr of Buffers versus Estimated IOs for Sequential Bottom Level access
 -------------        -------------
      1  - 20                  115
```

Print Space

```
PRINT SPACE FOR AREA PED.FFER011X-AREA FULL;
                                    AVAILABLE Space Distribution Report
                                              AVAIL         NUMBER
                                              SPACE        OF PAGES
AREA       PED.FFER011X-AREA
 PAGE SIZE            11,476
 PAGE RESERVE            400
    PAGES          2,150,001 THRU      2,151,000
                                              91-100%            884
                                              81-90 %              0
```

71-80 %	0
61-70 %	0
51-60 %	0
41-50 %	0
31-40 %	0
21-30 %	113
11-20 %	0
00-10 %	2
SMPS	1
TOTAL	1,000

FILE PED.FFER011X
BLOCKS 1 THRU 1,000

Total Space Allocated		11,476,000	
Total Space Available	(Percent)	10,456,812	(91.11%)
Total Reserved Space Available	(Percent)	399,192	(3.47%)
Total Unreserved Space Available	(Percent)	10,057,620	(87.64%)
Total Space Used		1,019,188	
Logically Full Pages		0	
Total Space Unusable	(Percent)	0	(0.00%)

AREA PED.FFER011X-AREA Distribution of USED Space Report

Record Type	Maximum Length	Occurrences	Total Space Used	Percent of Total Used
SR7	40	1	40	0.00
SR8	2,812	347	975,704	95.73
Space Inv.	11,444	1	11,444	1.12
Overhead	32	1,000	32,000	3.13

*** NO logically deleted records found ***

Index Calculator

Use the Index Calculator below to work out INDEX BLOCK CONTAINS and PAGE RESERVE clause

ALWAYS be certain all pages in an area CAN get to be over 70% full when storing SR8 records to avoid the "SMP I/O thrashing"

Input:	
Index Name	**SYS-PEPN-ID**
# Index Entries	**22,315**
Key Length	**31**
# Index Levels	3
Page Size	11,476
Device Type	3390

Output	
# Entries/SR8	**80**
Size of SR8	3,199
# SR8 (Level 0)	279
# SR8 (Level 1)	4
# SR8 (Level 2)	1
# SR8 (Level 3)	-
# SR8 (Level 4)	-
# SR8 (Level 5)	-
# Bytes for Level 0 SR8	892,521
# Bytes for Non Level 0 SR8	908,516
Total Bytes	1,801,037
Page Size	**4**
# SR8 Displacement Pages	**3**
# Non Displaced SR8s	5
# Pages	96
# Tracks	**25**

Input:	
Index Name	**SYS-PEPN-ID**
# Index Entries	**22,315**
Key Length	**31**
# Index Levels	3
Page Size	11,476
Device Type	3390

Output	
# Entries/SR8	**70**
Size of SR8	2,809
# SR8 (Level 0)	319
# SR8 (Level 1)	5
# SR8 (Level 2)	1
# SR8 (Level 3)	-
# SR8 (Level 4)	-
# SR8 (Level 5)	-
# Bytes for Level 0 SR8	896,071
# Bytes for Non Level 0 SR8	912,925
Total Bytes	1,808,996
Page Size	**4**
# SR8 Displacement Pages	**3**
# Non Displaced SR8s	6
# Pages	83
# Tracks	**21**