'Best Practices' for Records-Per-Block Settings

October, 2005

PROGRESS SOFTWARE

Table of Contents

Table of Tables

Table of Figures

Introduction

In order to configure the records-per-block parameter for optimal performance one must determine the proper setting for the application and implementation.

The purpose of this paper is to provide the reader with an understanding of how to properly set records-per-block with the goal of providing optimal performance and to avoid costly system crashes and possible data corruption.

OpenEdge RDBMS rowids are currently 32 bit signed integers. This means that there are 31 bits of values available for use in identifying records. A 1 bit sign is used to indicate a delete placeholder in unique indexes and to reserve recids for deleted records until the associated transaction commits.

Within these 31 bits, they are splitting them up into a block number and a row (record) within block number.

The number of records specified per block dictates how many bits are reserved to identify the row number within the block. The remaining bits are reserved to address the block number.

Figure 1 provides a binary representation of a rowid with the sign value *

Figure 1 – rowed with the sign value *

```
*000 0000 0000 0000 0000 0000 0000 0000
```

When a user specifies 256 rows-per-block (RPB) the database reserves 8 bits to identify each possible record in the block (0-255). This is true even if you don't use that many rows in each block. Due to block and record overhead, requesting 256 records per block means that the mean user data size is 10 bytes per record. This situation is a rare one. All the rowids for this area are therefore setup as depicted in figure 2.

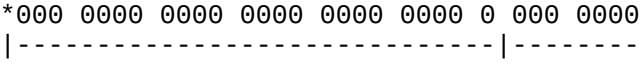
Figure 2 – 256 RPB

```
*000 0000 0000 0000 0000 0000 0000 0000
|----------------------------|---------|
  Addressable block portion    256 RPB
```

With this setup, the database can only address 23 bits of blocks. That nets out to 8,388,607 blocks.
8,388,607 blocks times the block size of 8192 = 68,719,468,544 = 64 GB of data.

If 128 RPB is specified then the database only needs 7 bits to represent the requested number of records-per-block leaving one more bit for the number of blocks that may be addressed. Following this configuration the rowids would break down as in figure 3.

Figure 3 – 128 RPB



Using the configuration above 24 bits of blocks may be addressed. This nets out to 16,777,215 blocks.
16,777,215 time the block size of 8192 = 137,438,953,472 = 128 GB of data.

The examples above clearly illustrate that properly sizing the RPB is extremely important.

If the records-per-block is set too low, then the data blocks may contain substantial free space that is undesirable as it wastes space in the buffer pool and on disk and could increase the number of database I/Os per records read.

Setting records-per-block too high may artificially limit the amount of addressable space in a single storage area and may cause expensive record fragmentation depending on how the application inserts and updates records.

The way to fix an improperly set record-per-block value is by performing a costly data movement operation such as binary dump and load, tablemove, or data dump/load/bulkload w/index rebuild).

Properly setting records-per-blocks is not a difficult task and is based on average record size. An effective approach is to take the blocksize / avg record size (from tabanalys) and set it to the next higher RPB value.

A more complex approach is also available. This approach should yield better record packing while allowing the most addressable data and the least record fragmentation. The tables below provide insight into this approach. Note the differences between v9 and OpenEdge 10 with Type I storage areas and OpenEdge 10 with Type II storage areas. All values are based on 8K block size

Table 1 – v9 RPB calculation

RPB: Records-Per-Block Setting
AUD: Average User Data (vs Average Record Size from tabanalysis) - AUD is used for new database planning; ARS from tabanalys is used for existing database reconfiguration.

V9 Type I storage area 8K database block size (create,toss) limits of (600, 2000)

Addressable data	RPB	AUD per record	Average Record size (from tabanalys)
64 GB	256:	10 bytes	27
128 GB	128:	40 bytes	57
256 GB	64:	99 bytes	116
512 GB	32:	217 bytes	234
1024 GB	16:	452 bytes	473
2048 GB	8:	925 bytes	946
4096 GB	4:	1872 bytes	1893
8192 GB	2:	3765 bytes	3786
16384 GB	1:	8151 bytes	8170

The amount of addressable data does not change between V9 and OpenEdge 10. Also note that the theoretical number of addressable records per area does not change between RPB settings if set properly.

Table 2 – OpenEdge 10 with Type II storage areas RPB calculation

V10 Type II storage area 8K database block size with the default (Create,Toss) limits of (150,300)

RPB	AUD per record	Average Record size (from tabanalys)
256:	12 bytes	29
128:	43 bytes	60
64:	105 bytes	122
32:	229 bytes	246
16:	476 bytes	495
8:	973 bytes	992
4:	1968 bytes	1987
2:	3958 bytes	3977
1:	8101 bytes	8120

Table 3 – OpenEdge 10 with Type I storage areas RPB calculation

V10 Type I storage area 8K database block size with the default (Create,Toss) limits of (150,300)

RPB	AUD per record	Average Record size (from tabanalys)
256:	12 bytes	29
128:	43 bytes	60
64:	106 bytes	123
32:	231 bytes	248
16:	480 bytes	499
8:	981 bytes	1000
4:	1984 bytes	2003
2:	3990 bytes	4009
1:	8151 bytes	8170

Appendix A provides detail on how the values above were calculated. This may prove useful to those who are interested in calculating their own data.

Values are calculated based on size, not by ultimate size from an insert followed by one or more updates.

Appendix A

For those interested in calculating their own data, there is additional detail below on how the values in tables 1, 2, and 3 were calculated. The numbers are calculated based on insertion size, not by ultimate size from an insert followed by one or more updates.

For V9:
Use ‘create limit’ instead of ‘toss limit’ with the assumption that all the records are just being created as is the case in a dump and load (no updates). The CreateLimit is applied to the algorithm rather than the TossLimit. The record version overhead increased from 12 to 14 as the vector indicator and the end of array marker which combined adds up to 2 byes more overhead per record.

Note: The described behavior is subject to change without notice.

Each Block:
    BlockSize(BS) 8192
    Records-Per-Block(RPB) 256
    Create Limit (CL) 600
    Toss Limit(TL) 2000
    Block header size(BHS) 16
    Record block header overhead(RBO) 4
Each Record:
    Record Offset(RO) 2
    Record Length (RL) 2
    Record Versioning overhead(RVO) 14
Each Field of Each Record:
    Field length(FL) 1
    Number of Filed (NF) 1

There are additional items affecting size such as field sizes > 256, array fields, records with >16 fields as there is a skip table that adds more space per record, and table numbers > 255.

Space Available for records (SAFR) = BS - CL - BHS - RBO = 7572 bytes
Avg Size of record (ASOR) = SAFR / RPB ~= 29

Avail User Data (AUD) = ASOR - RO - RL - RVO - (FL * NF) = 10 bytes max available user data in record creation of 1 field
12 bytes of user data for update after all records in the block were created or the block would be removed from the rm chain even sooner and you couldn't get 256 records at all.

NOTE: This value is the amount of actual user data. The value gotten from tabanalys includes user data as well as RVO and FL)

RPB	AUD per record	Average Record size (from tabanalys)
256:	10 bytes	27
128:	40 bytes	57
64:	99 bytes	116
32:	217 bytes	234

(for fields > 256 length takes 3 bytes, not 1 so subtracting two more bytes)

16:	452 bytes	473

```
  8:  925 bytes        946
  4: 1872 bytes       1893
* 2: 3765 bytes       3786
* 1: 8151 bytes       8170
```

For OpenEdge 10:

In OpenEdge 10 the CreateLimit and TossLimit have been decreased that allows for tighter packing of data but has an effect on larger records.

The described behavior is subject to change without notice.

```
Each Block:
    BlockSize(BS) 8192
    Records Per Block(RPB) 256
    Create Limit (CL) 150
    Toss Limit(TL) 300
    Block header size(BHS) 16 Type I, 64 or 80 Type II
    Record block header overhead(RBO) 4
Each Record:
    Record Offset(RO) 2
    Record Length (RL) 2
    Record Versioning overhead(RVO) 14
Each Field of Each Record:
    Field length(FL) 1
    Number of Fields (NF) 1
```

There are additional things affecting size such as filed sizes > 256, array fields, records w/>16 fields as there is a skip table
That adds more space per record, and table numbers > 255.

OpenEdge 10 Type II 8K db w/create/toss limits 150/300 (and assuming 80 byte block header which is not always true).

Space Available for records (SAFR) = BS - CL - BHS - RBO = 7958 bytes

Avg Size of record (ASOR) = SAFR / RPB ~= 31

Avail User Data (AUD) = ASOR - RO - RL - RVO - (FL * NF) = 12 bytes max available user data in record creation of 1 field
In OpenEdge 10, average of only 0.5 additional bytes of user data for update after all records in the block were created or the block would be removed from the rm \chain even sooner and you couldn't get 256 records at all.

NOTE: This value is the amount of actual user data. The value gotten from tabanalys includes user data as well as RVO, FL and RL)

```
RPB   AUD per record  Average Record size (from tabanalys)
256:   12 bytes         29
128:   43 bytes         60
 64:  105 bytes        122
 32:  229 bytes        246
```

(for fields > 256 length takes 3 bytes, not 1 so subtracting two more bytes)

```
*16:  476 bytes        495
* 8:  973 bytes        992
```

```
* 4: 1968 bytes       1987
* 2: 3958 bytes       3977
* 1: 8101 bytes       8120


V10 Type I 8K db w/create/toss limits 150/300

Space Available for records (SAFR) = BS - CL - BHS - RBO = 8022 bytes

RPB   AUD per record  Average Record size (from tabanalys)
256:   12 bytes         29
128:   43 bytes         60
 64:  106 bytes        123
 32:  231 bytes        248
(for fields > 256 length takes 3 bytes, not 1 so subtracting two more
bytes)
 16:  480 bytes        499
  8:  981 bytes       1000
  4: 1984 bytes       2003
* 2: 3990 bytes       4009
* 1: 8151 bytes       8170
```

Corporate and North American Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters
Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters
Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters
Progress Software Pty. Ltd., 1911 Malvern Road, Malvern East, 3145, Australia Tel: 61 39 885 0544 Fax: 61 39 885 9473

PROGRESS SOFTWARE

www.progress.com