Dr. Stephan Bühne
Principal Technical Consultant
Oracle SAP Solution Center Walldorf

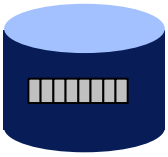
ORACLE

Oracle Reorganisation

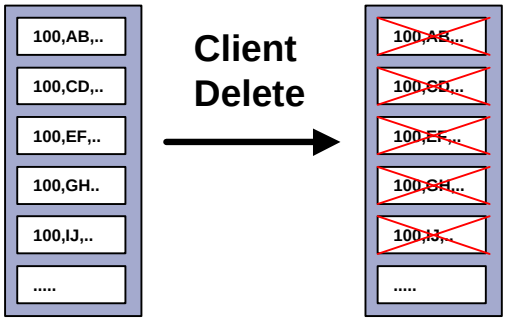
ORACLE

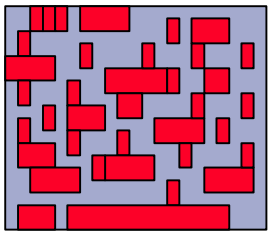
- Why to reorganize
- Traditional Space Management
- New Automatic Space Mangement
- Locally Managed Tablespaces
- Using Parallel Processing

Why to Reorganise



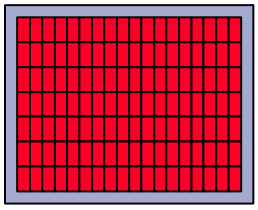
Free Up
Disk Space



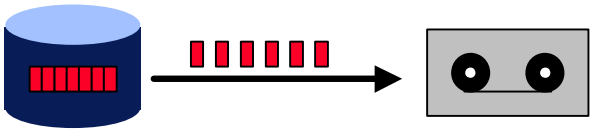


Space Fragmentation
in Tablespace

?



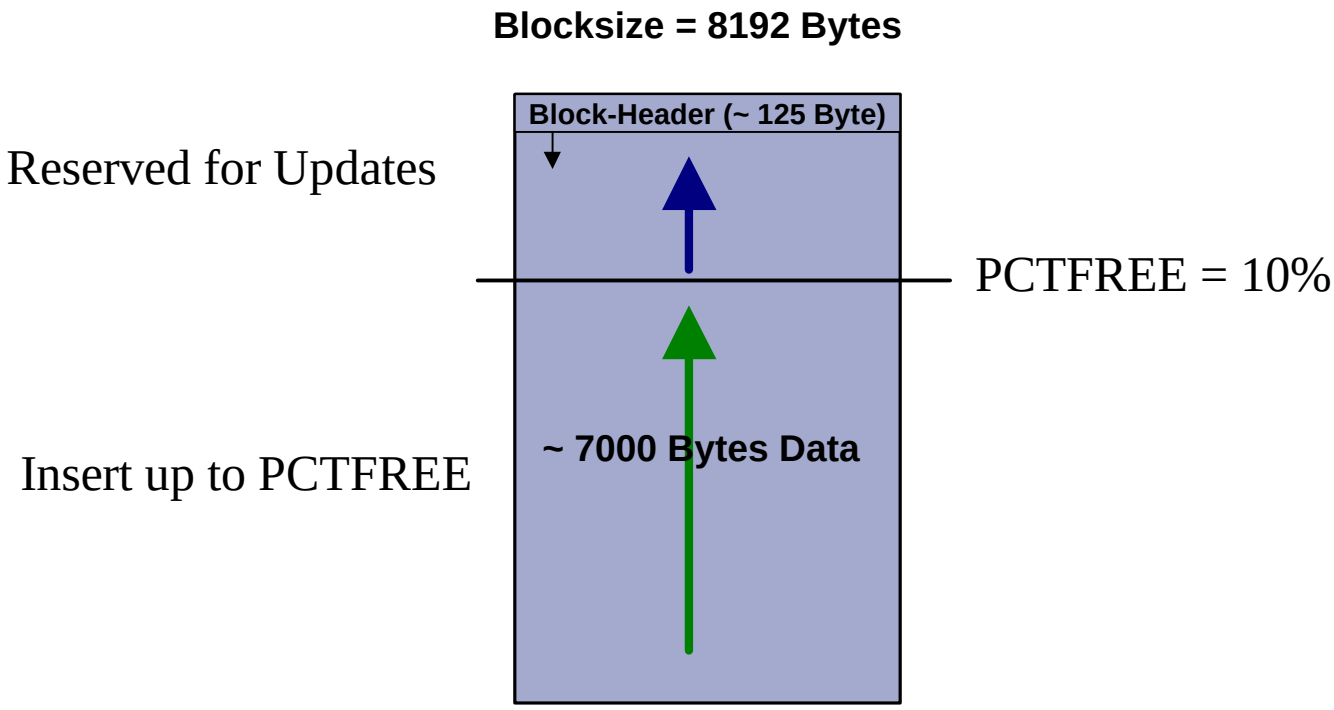
Number of Extents



Archiving

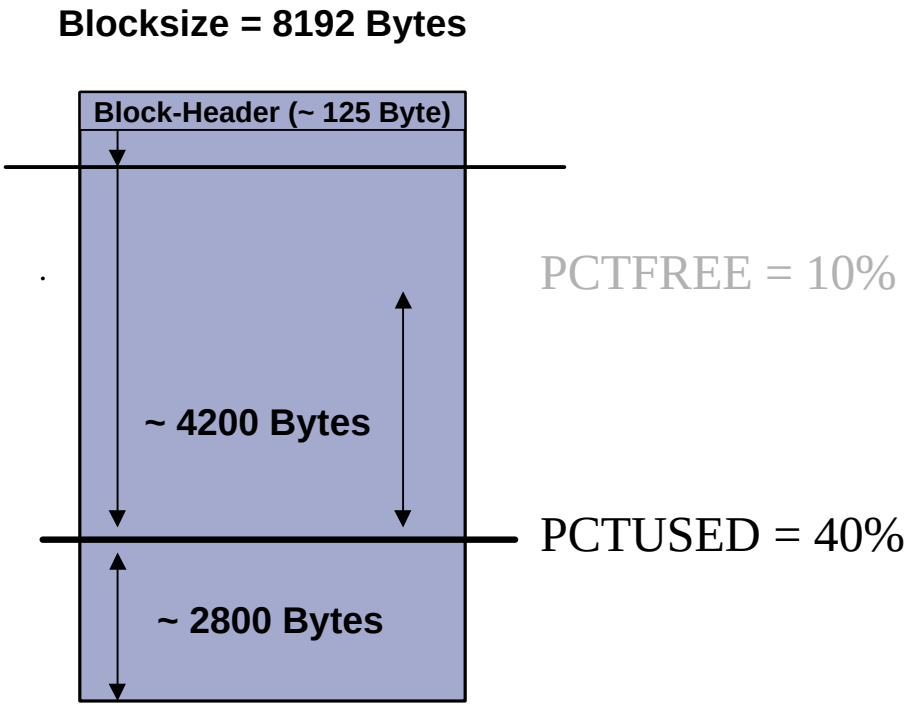
- Why to reorganize
- **Traditional Space Management**
- New Automatic Space Mangement
- Locally Managed Tablespaces
- Using Parallel Processing

Space Management Oracle: Insert/ Update



Space Management Oracle: Delete

Current Block Filling Level

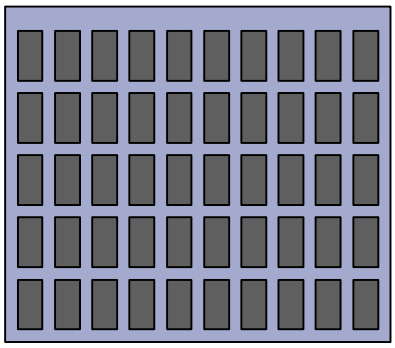


Archiving Procedure

DELETE FROM EDIDC WHERE CREDAT < 01.01.2003

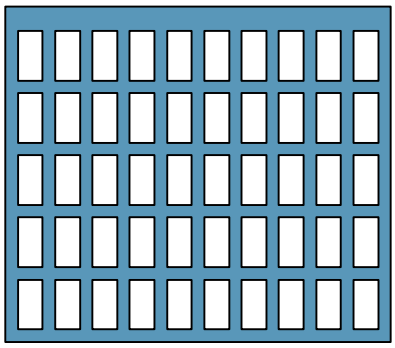
DELETE FROM EDIDC WHERE CREDAT < 01.01.2003 and WERKS = 001

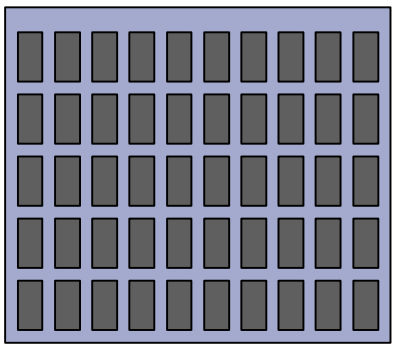
Influence on Data distribution



DELETE FROM EDIDC WHERE

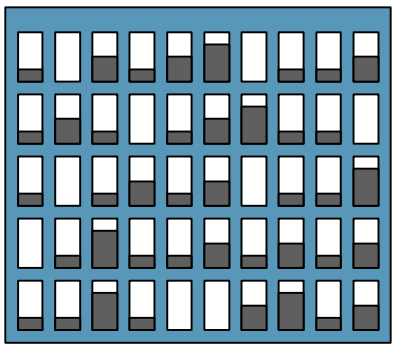
CREDAT < 01.01.2003





DELETE FROM EDIDC WHERE

CREDAT < 01.01.2003
AND WERKS = 001

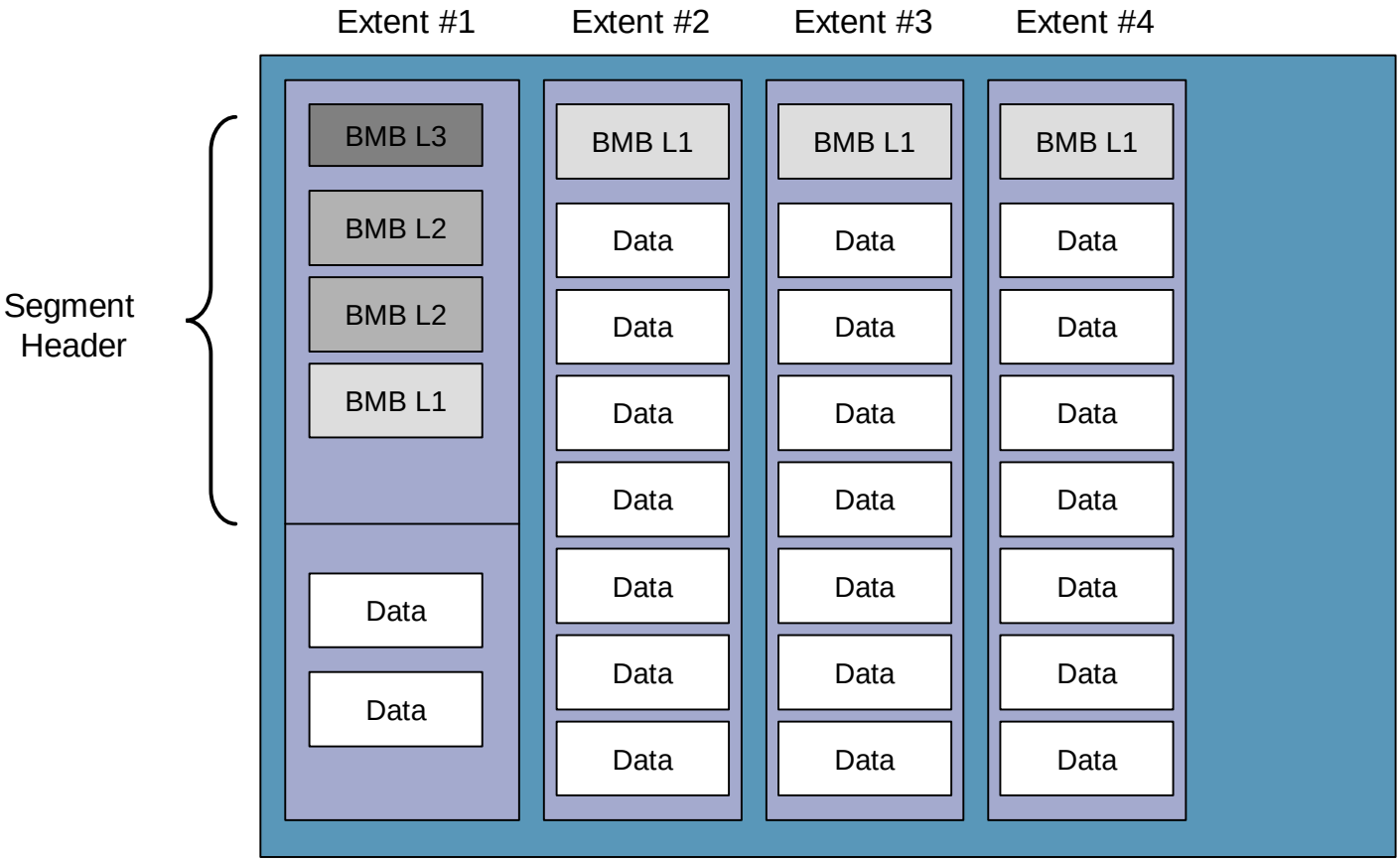


- Why to reorganize
- Traditional Space Management
- **New Automatic Space Mangement**
- Locally Managed Tablespaces
- Using Parallel Processing

Automatic Space Management: Features

- **Ease of Use**
  => No PCTUSED, FREELISTS, FREELISTS
- **Better space utilization**
  => Freed space is reused immediatly
- **Better concurrency handling**
  => No contention on parallel processing
- **Better performance**
  => Faster seek operation for free space

Automatic Segment Space Management



BMB: Bit-Mapped Block

BMB: Internal Structure

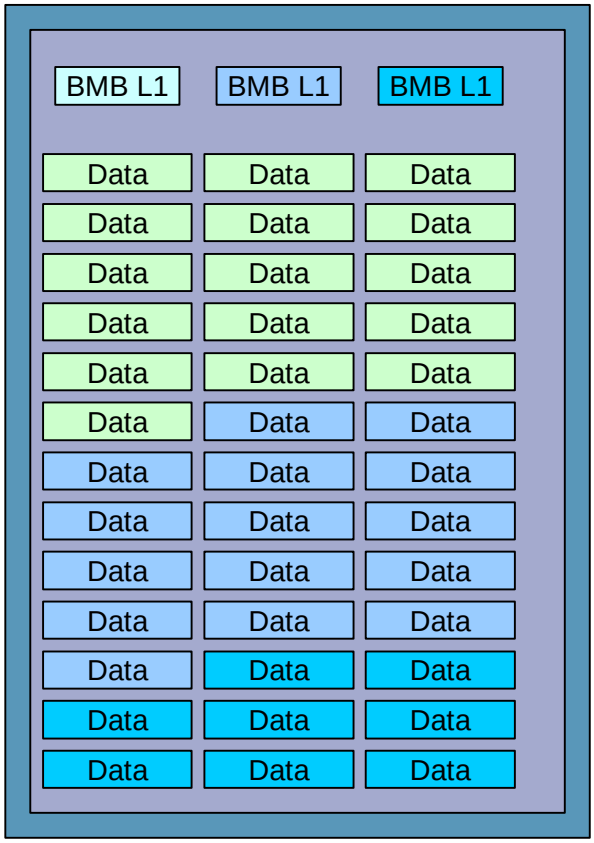
```
Freeness Status:  nf1 1     nf2 1     nf3 0     nf4 11    First free datablock : 3
  Bitmap block lock opcode 0
  Locker xid:     :  0x0000.000.00000000
    Highwater::  0x01000021  ext#: 0     blk#: 16     ext size: 16
 #blocks in seg. hdr's freelists: 0
 #blocks below: 13
 mapblk  0x00000000  offset: 0
 HWM Flag: HWM Set
 ------------------------------------------------------------
 DBA Ranges :
 ------------------------------------------------------------
  0x01000011  Length: 16     Offset: 0
  0:Metadata   1:Metadata   2:Metadata   3:75-100% free
  4:75-100% free   5:75-100% free   6:75-100% free   7:75-100% free
  8:75-100% free   9:75-100% free   10:75-100% free   11:0-25% free
  12:25-50% free   13:75-100% free   14:75-100% free   15:75-100% free
```

nf1 = block is 0-25% free
nf2 = block is 25-50% free
nf3 = block is 50-75% free
nf4 = block is 75-100% free

Automatic Segment Space Management

Extent

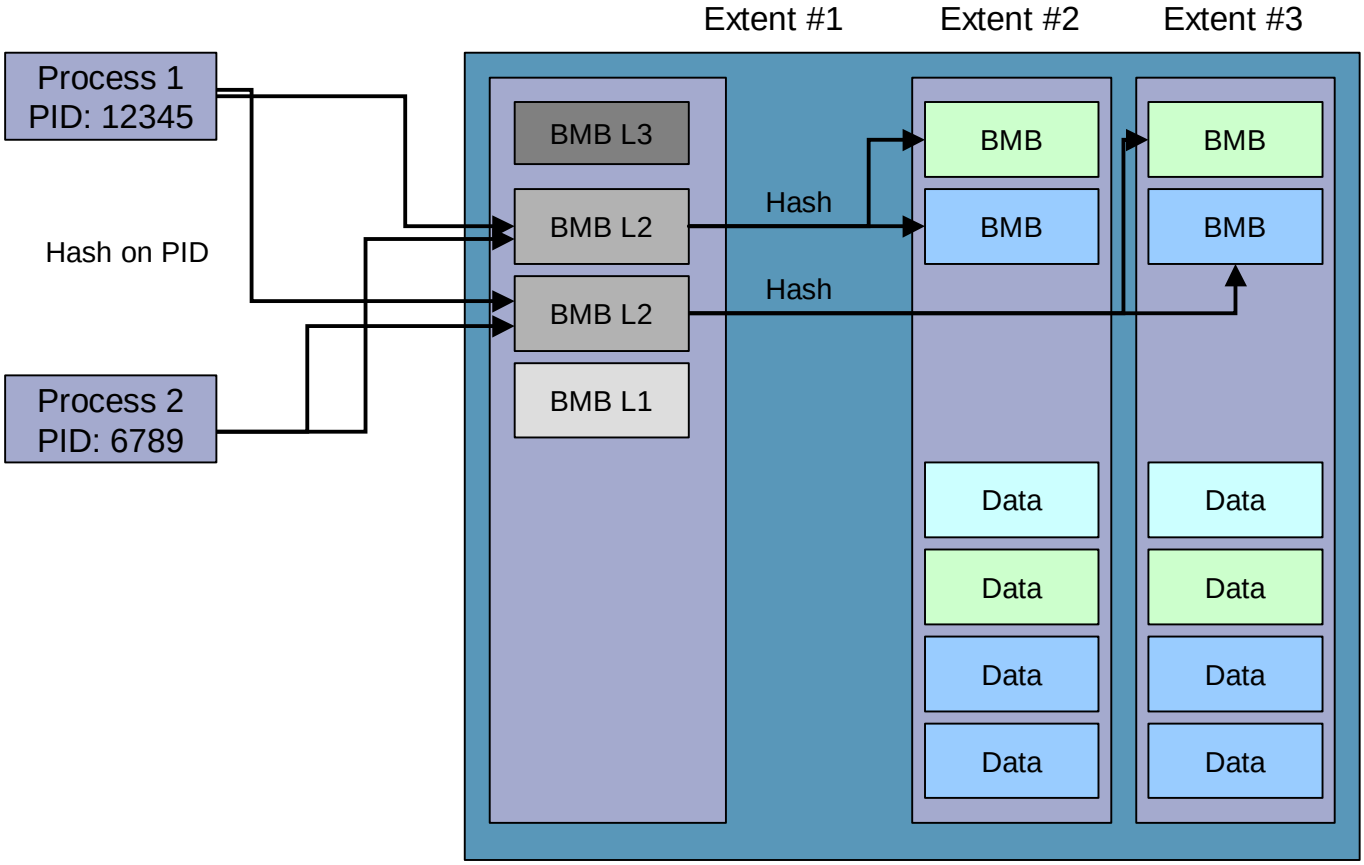
BMB: Bit-Mapped Block



Each BMB controls up to:

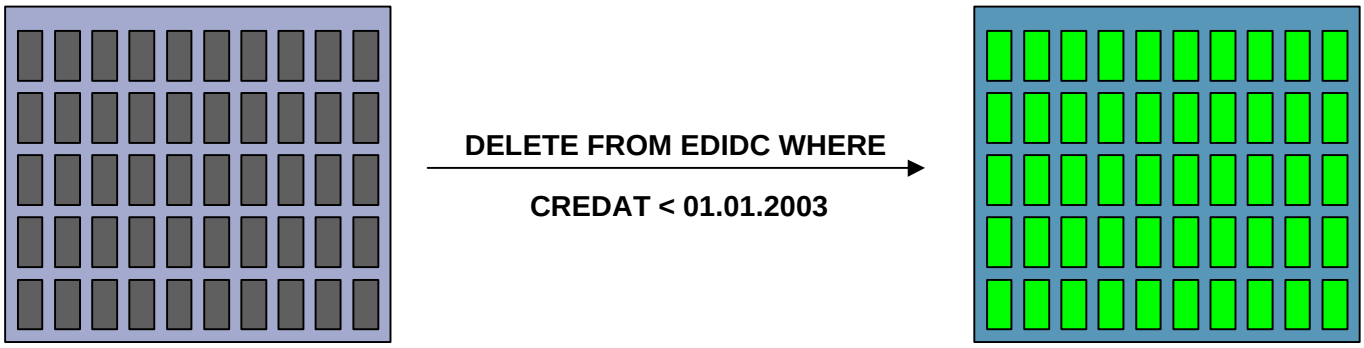
=> 16 blocks until the segment is 1Mb
=> 64 blocks until the segment is 32Mb
=> 256 blocks until the segment is 1Gb
=> 1024 blocks once segment is >1Gb

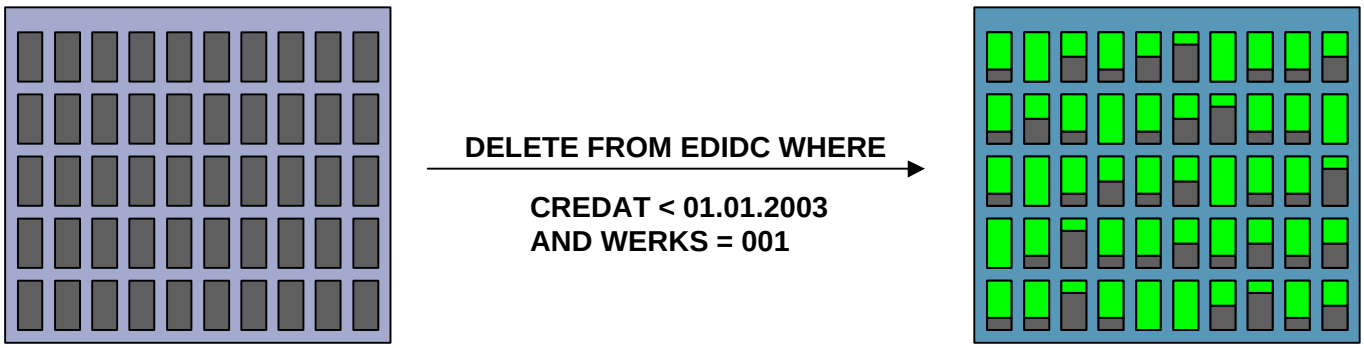
Automatic Segment Space Management



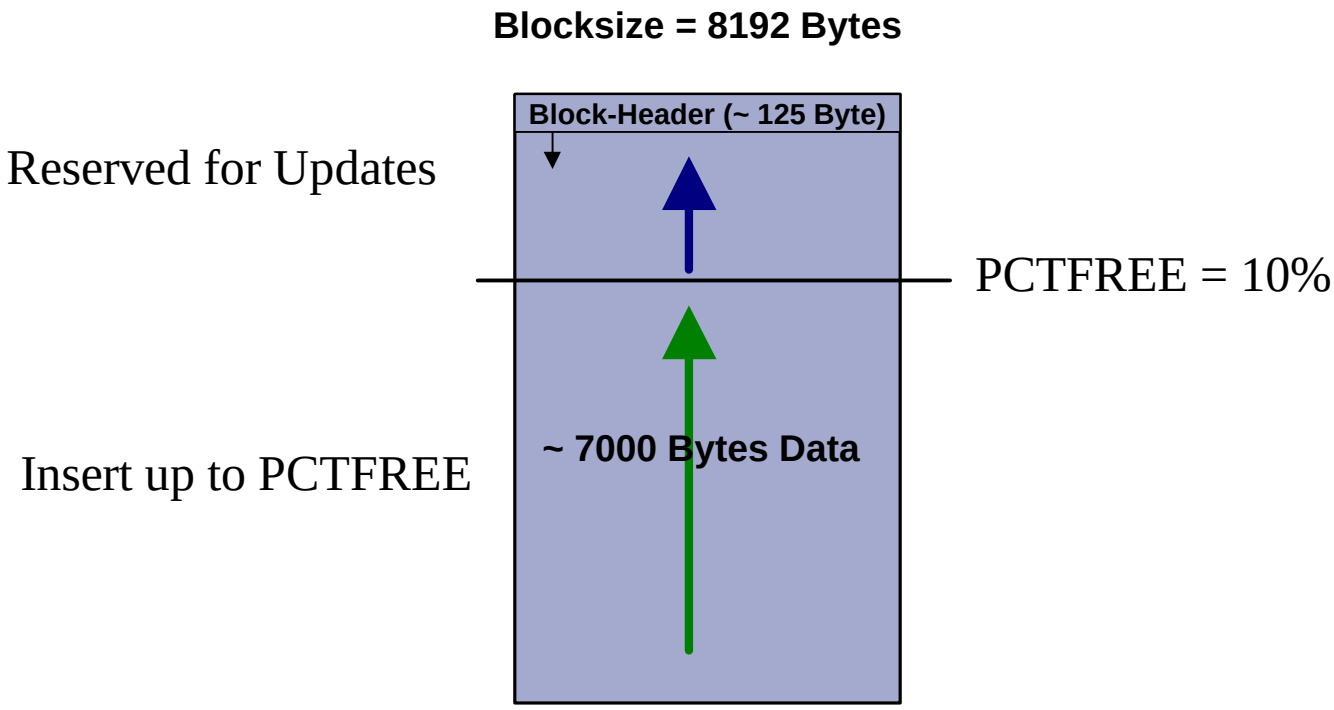
BMB: Bit-Mapped Block

Influence on Data distribution



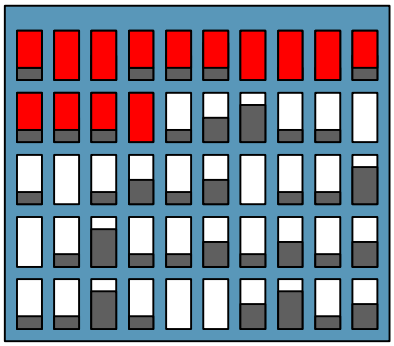


Space Management Oracle: Insert/ Update



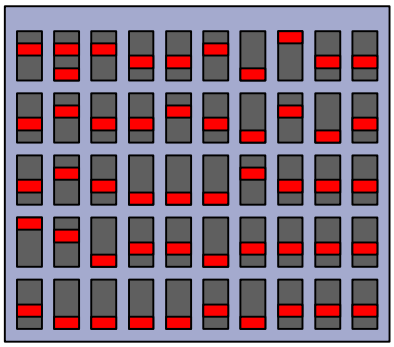
Influence on Data distribution

Performance ?



or

Space ?



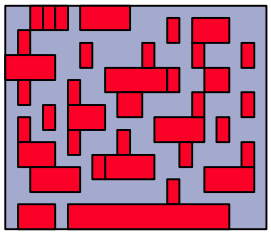
Summary Automatic Space Management

- Improved performance for parallel load
- Less administration
- Easy to reuse space
- Increased Data distribution on blocks

- Why to reorganize
- Traditional Space Management
- New Automatic Space Mangement
- **Locally Managed Tablespaces**
- Using Parallel Processing

Locally Managed Tablespaces

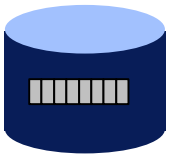
Dictionary Managed



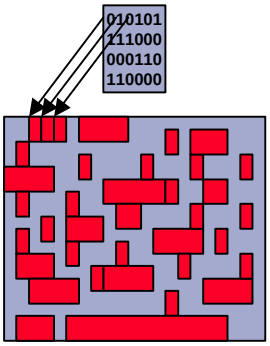
Insert, Update, Delete

FET$, UET$

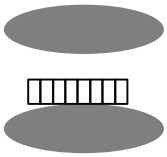
System Tablespace



Locally Managed

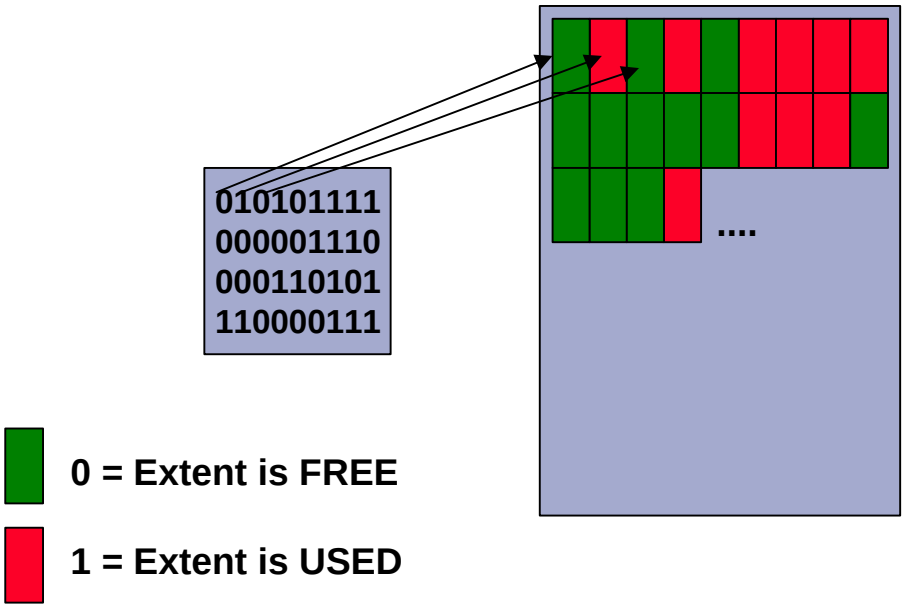


System Tablespace



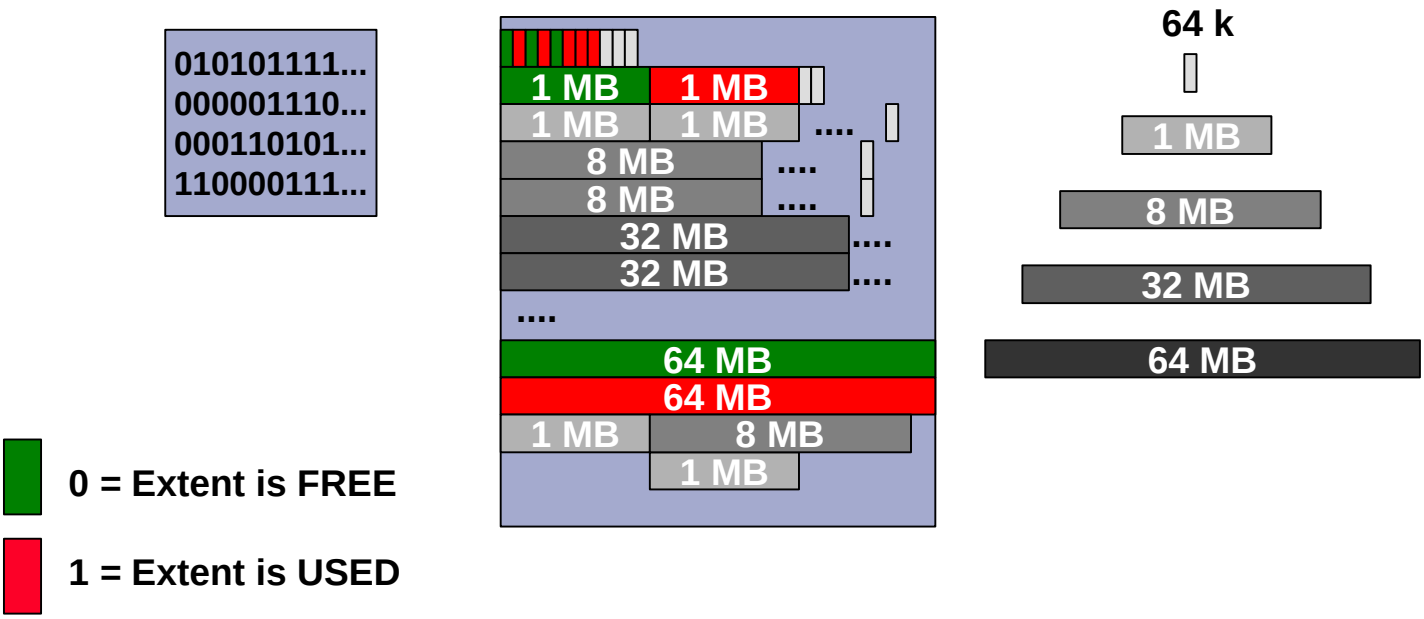
Internal Structure of Locally Managed TBS

Uniform Size: All Extents have the same size



Internal Structure of Locally Managed TBS

Autoallocate: 5 different extent sizes will be used



Advantages of Locally Managed Tablespaces

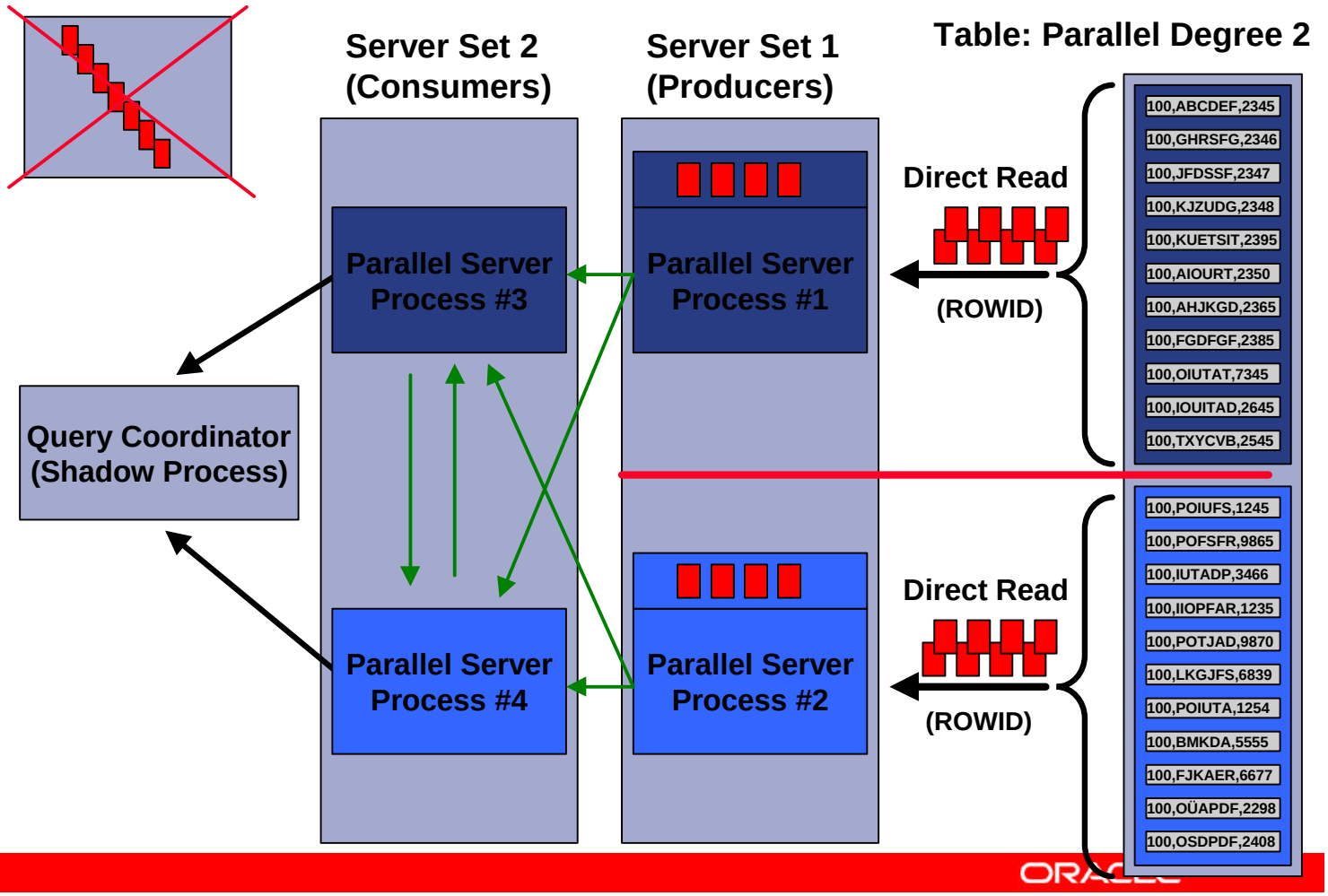
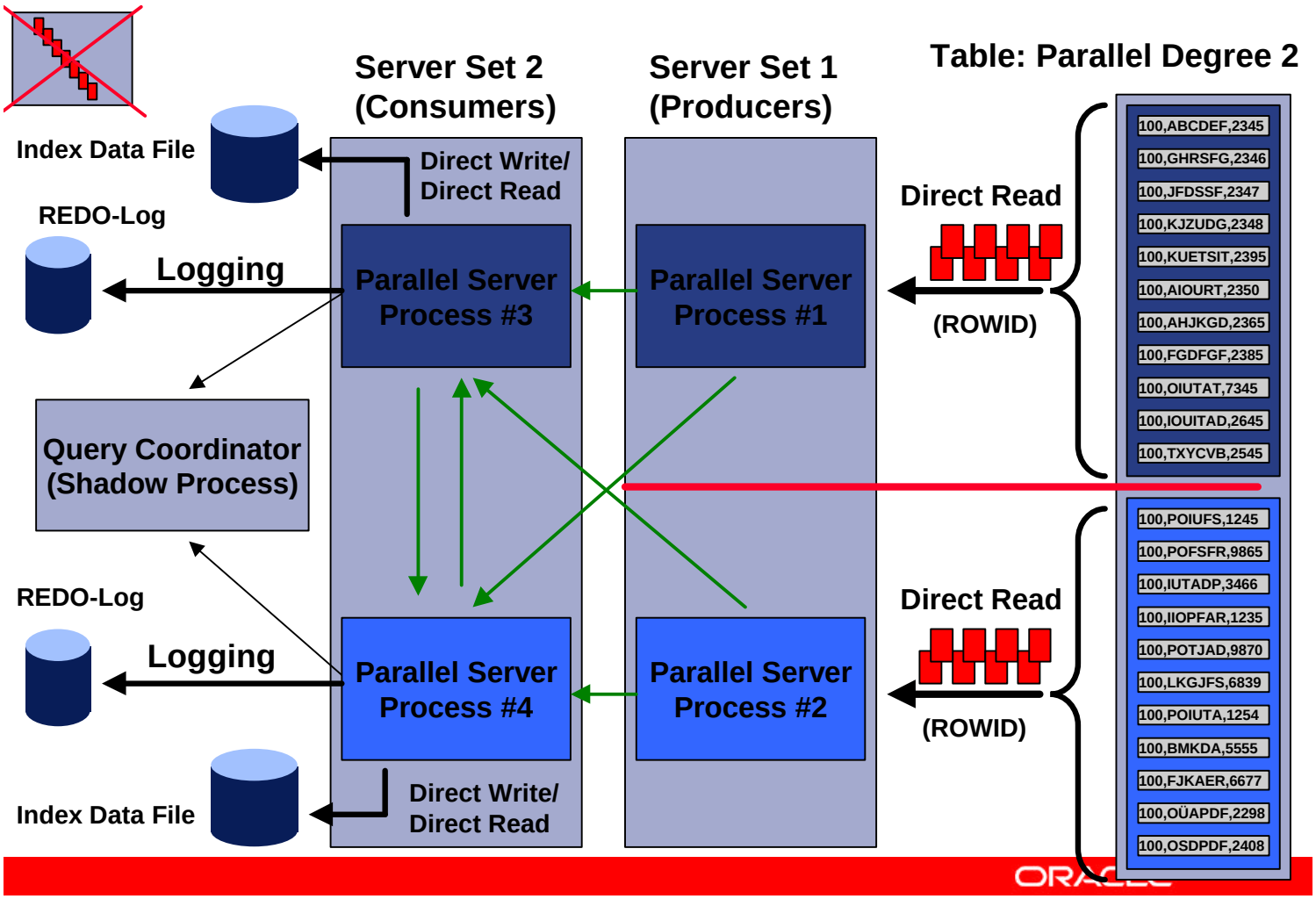
- No Initial or Next Extent Definition necessary
- No space fragmentation, forced by very different extent sizes (16K – 2GB)
- Better usage of available space within the tablespace
- No Extent limitation
- Number of Extents has no impact on the system performance
- Better performance during insert/ delete operations and parallel load

Reorganisation Methods

- Export/ Import
- Create Table As Select (CTAS)
- Online-Index Reorganisation
- Online-Table Reorganisation (Oracle 9i only)

- Why to reorganize
- Traditional Space Management
- New Automatic Space Mangement
- Locally Managed Tablespaces
- **Using Parallel Processing**

Parallel Query: FTS & Sorting



Parallel Query: Create Index