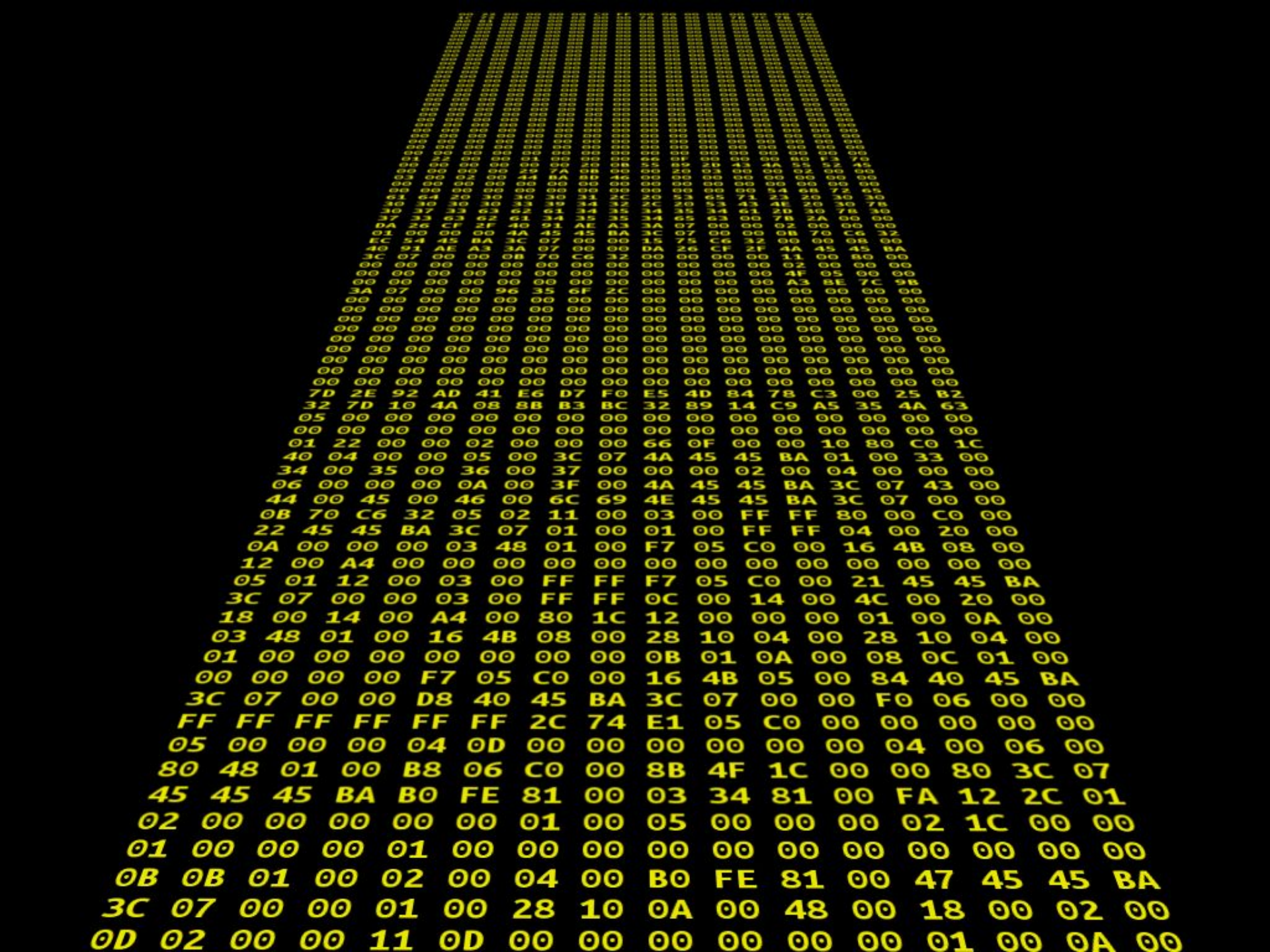
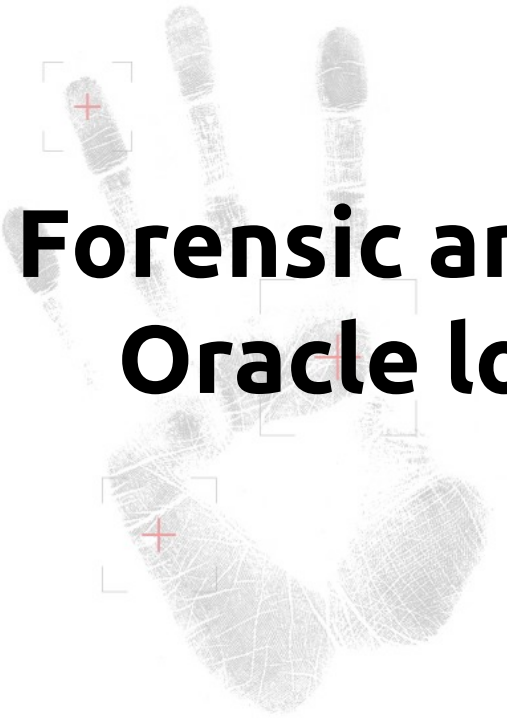


ORACLE Gold Partner



Forensic analysis of Oracle log files

Jure Kajzer
Abakus PLUS d.o.o.

19. Strokovno srečanje
SIOUG 2014
13.-14. oktober 2013

Abakus plus d.o.o.

ORACLE Gold Partner

History

from 1992, ~20 employees

Applications:

special (DB – Newspaper Distribution, FIS – Flight Information System)
ARBITER – the ultimate tool in audit trailing
APPM - Abakus Plus Performance and Monitoring Tool

Services:

DBA, OS administration , programming (MediaWiki, Oracle)
networks (services, VPN, QoS, security)
open source, monitoring (Nagios, OCS, Wiki)

Hardware:

servers, **backup server**, **SAN storage**, firewalls

Infrastructure:

from 1995 GNU/Linux **(19 years of experience !)**
Oracle on GNU/Linux: since RDBMS 7.1.5 & Forms 3.0 **(before Oracle !)**
>20 years of experience with High-Availability!

Sava
Gorenjska Banka – Banka s poslubom
Aerodrom Ljubljana
Mercator
GOODYEAR
BANKA SLOVENIJE – EVROSISTEM
Mestna občina Ljubljana
MESTNA OBČINA KOPER – COMUNE CITTA DI CAPODISTRIA
futuraplus
Iskra – Iskra MIS
DELO PRODAJA
KONTROLA ZRAČNEGA PROMETA SLOVENIJE

GENERALI Zavarovalnica

Aerodrom Ljubljana

BANKA SLOVENIJE
BANK OF SLOVENIA
EUROSYSTEM

DELO PRODAJA

KONTROLA ZRAČNEGA PROMETA SLOVENIJE

Gorenjska Banka
Vse, kar šteje.

GOODYEAR

HRANILNICA LON
Bančništvo na Ljubezniv Oseben Način

Mestna občina Ljubljana

Mercator

NLB Vita
Življenjska zavarovalnica

MESTNA OBČINA KOPER
COMUNE CITTÀ DI CAPODISTRIA

Sava

Abakus
As na disku.

What are you on about you muppet?

Existing options

… and why not use them

- LogMiner
- ALTER SYSTEM DUMP LOGFILE

Why not?

✗ requires a working (compatible) database
✗ susceptible to log corruption
✗ needs all logs since last dictionary (LogMiner)

Main structures

- Block
  - static size (defined in block 0)
  - defaults to disk block size
  - endianess

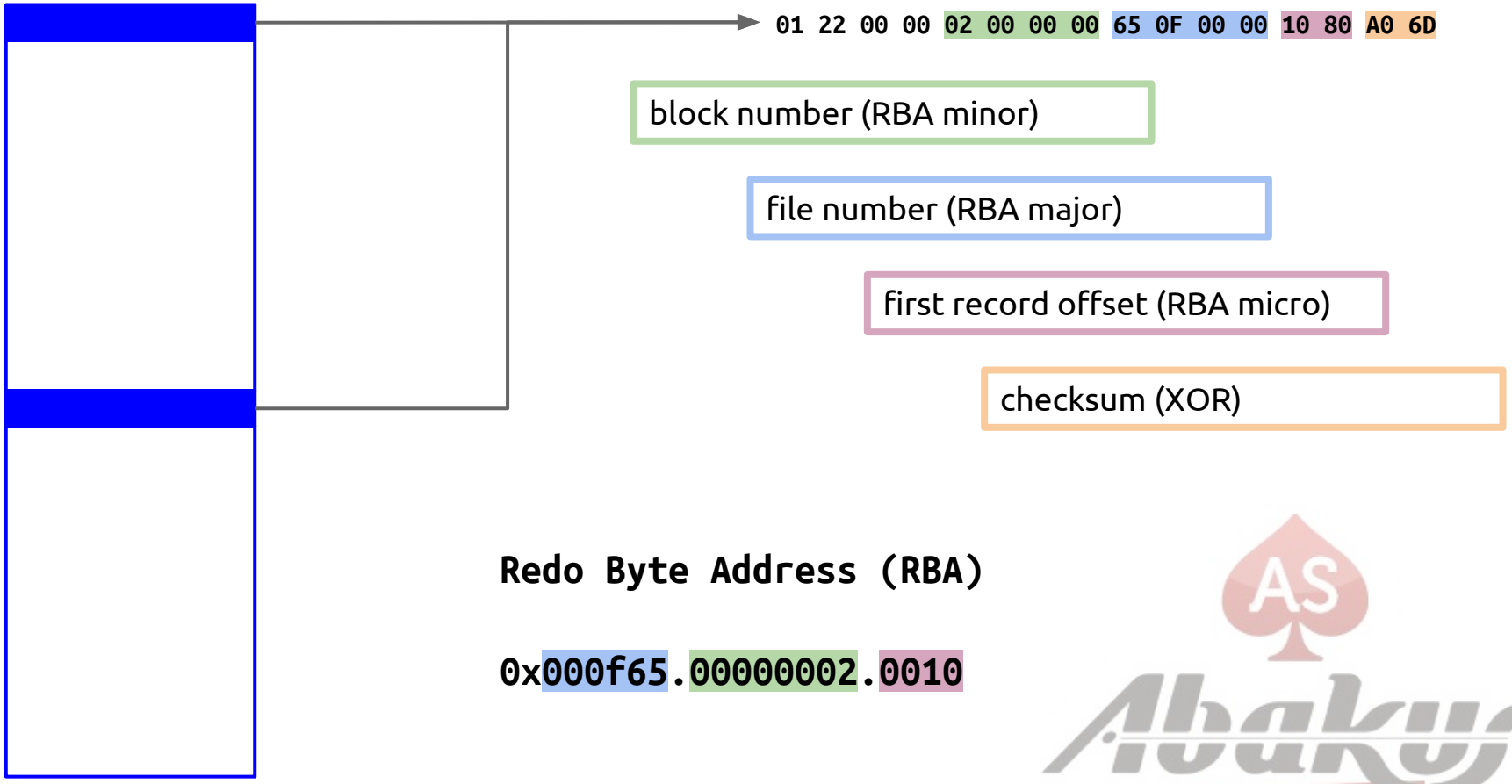
```
00 22 00 00 00 00 C0 FF 00 00 00 00 00 00 00 00
1B 58 00 00 00 02 00 00 7D 00 00 00 7D 7C 7B 7A
A0 81 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- Record
  - vector (size followed by data)
  - chained

- Change
  - static header size
  - vector of data part sizes
  - data parts

Block

Structure



Redo Byte Address (RBA)

0x000f65.00000002.0010

Block 0x01 (Log header)

```
01 22 00 00 01 00 00 00 65 0F 00 00 00 80 83 32
00 00 00 00 00 00 20 0B 55 89 2D 43 4A 55 52 45
00 00 00 00 1E 7A 0B 00 00 20 03 00 00 02 00 00
02 00 02 00 44 BA 8D 46 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 54 68 72 65
61 64 20 30 30 30 31 2C 20 53 65 71 23 20 30 30
30 30 30 30 33 39 34 31 2C 20 53 43 4E 20 30 78
30 37 33 63 62 61 34 35 34 35 30 63 2D 30 78 30
37 33 63 62 61 34 35 34 35 34 61 00 7E 00 00 00
DA 26 CF 2F 40 91 AE A3 3A 07 00 00 02 00 00 00
01 00 00 00 0C 45 45 BA 3C 07 00 00 F7 6F C6 32
4A 45 45 BA 3C 07 00 00 0B 70 C6 32 00 00 08 00
40 91 AE A3 3A 07 00 00 DA 26 CF 2F 0C 45 45 BA
3C 07 00 00 F7 6F C6 32 00 00 00 00 11 00 80 00
00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 26 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 A3 8E 7C 9B
3A 07 00 00 96 35 6F 2C 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
59 A6 DF 57 0D 95 86 AD 3D 70 3B D3 4D 40 2E 10
32 7D 10 4A 08 8B B3 BC 32 89 14 C9 A5 35 4A 63
05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

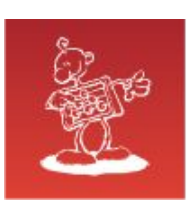
→ DB version: **11.0.2.0**
→ DB ID: **1127057749**
→ DB name: **JURE**
→ Activation ID: **1183693380**
→ Description: **Thread 0001, Seq# 0000003941, SCN 0x073cba45450c-0x073cba45454a**

→ Reset SCN: **7948435624256**
→ Thread: **1**

→ Low SCN: **7957404564822**
→ Low epoch: **07/02/2014 15:58:04**

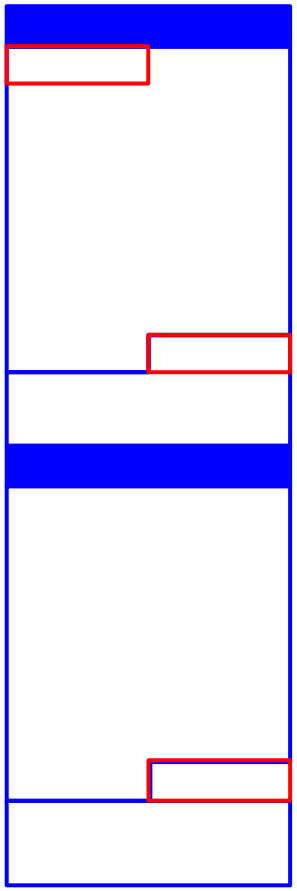
→ Next SCN: **7957404564977**
→ Next epoch: **07/02/2014 15:59:00**

→ ...



Record

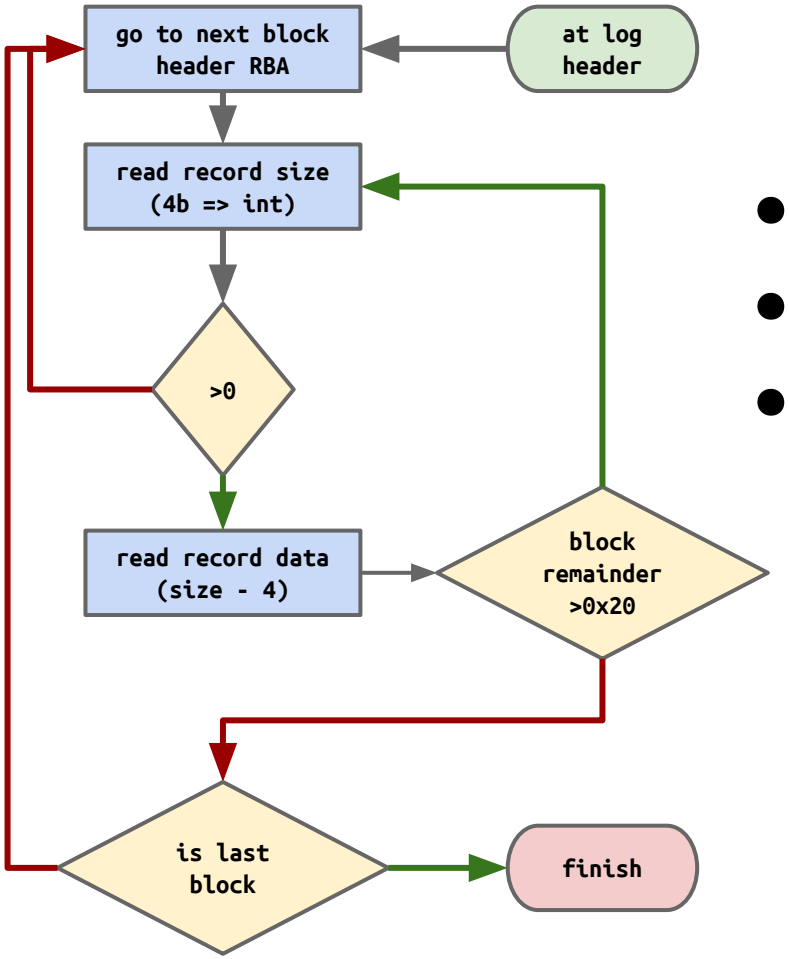
Structure



- Record size (4B)
- Record header (20B or 64B)
  - VLD byte
  - SCN
  - epoch (if VLD 4th bit set)
- Changes

Record

Chain



- Block header => first record RBA
- Record size 0 => last record in block
- Record header is never split

Chained record

```
0C 00 40 00 10 00 0C 00 0C 00 04 00 11 0D 45 BA
3C 07 00 00 0A 00 0E 00 F5 AB 07 00 7A 07 C0 00
03 24 03 00 00 00 00 00 00 00 00 00 70 38 FA 73
E3 2A 00 00 14 20 CF 77 02 01 01 00 0D 45 45 BA
3C 07 00 00 02 02 3C 07 09 45 45 BA 02 20 F5 00
20 00 00 00 02 00 00 00 F5 00 F6 00 05 C4 15 31
06 4D 05 C4 15 31 06 4E 00 81 FE 90 00 0A 00 81
FE 90 00 08 06 00 06 00 B4 00 00 00 01 2E 3C 07
0E 45 45 BA 02 00 57 49 44 2C 31 30 5D 00 00 00
04 06 01 00 02 00 04 00 E6 4B 82 00 0D 45 45 BA
3C 07 23 00 01 00 82 10 06 00 10 00 01 00 45 BA
01 22 00 00 03 00 00 00 65 0F 00 00 8C 80 06 73
0E 45 45 BA 3C 07 00 00 0A 00 0E 00 F5 AB 07 00
02 07 00 00 04 06 01 00 02 00 04 00 DC A2 81 00
0D 45 45 BA 3C 07 00 00 01 00 81 10 06 00 10 00
01 00 00 00 0E 45 45 BA 3C 07 00 00 0A 00 0E 00
F5 AB 07 00 02 03 00 00 04 06 01 00 02 00 04 00
90 FE 81 00 0D 45 45 BA 3C 07 00 00 01 00 23 10
06 00 10 00 01 00 0E 00 0E 45 45 BA 3C 07 00 00
0A 00 0E 00 F5 AB 07 00 02 00 28 00 2C 00 00 00
01 00 3C 07 0E 45 45 BA 02 00 00 00 07 00 02 00
08 00 06 00 05 02 11 00 03 00 FF FF 80 00 C0 00
03 45 45 BA 3C 07 00 00 2C 00 00 00 01 00 3C 07
0E 45 45 BA 02 00 00 00 07 00 02 00 08 00 06 00
05 02 11 00 03 00 FF FF 80 00 C0 00 03 45 45 BA
3C 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

RECORD CHAIN != ROW CHAIN

- → Record with size 0xB4 spans across two blocks
- → Block header only contains first block address
- → Record size == 0 signals no more records in block

Change

Structure

- Fixed sized header (24B)
  - operation code
  - OBJECT_ID / DATA_OBJECT_ID
- Data part sizes
  - vector of short values
  - aligned sizes (mind the gap)
- Data parts
  - types and order mostly depend on operation code

Change meta

```
05 01 24 00 03 00 FF FF 02 03 C0 00 9E 46 45 BA
3C 07 A2 00 11 00 FF FF 0C 00 14 00 18 00 10 00
14 00 14 00 5C 00 D4 18 22 00 5D 3E 0A 00 04 00
1D AC 07 00 04 24 12 00 04 11 04 00 04 11 04 00
01 00 00 00 00 00 00 00 0B 01 04 11 00 00 1D AC
02 0D 3C 07 E8 FA E1 B5 02 03 C0 00 04 24 10 00
F7 64 86 00 D2 39 81 00 FA 12 23 01 01 00 00 00
06 00 00 3E 02 1C 00 00 01 00 00 00 01 00 00 00
00 00 00 00 00 00 00 00 0B 02 01 00 02 00 04 00
F7 64 86 00 9E 46 45 BA 3C 07 00 00 06 00 04 11
5C 00 10 00 36 00 07 00 02 00 08 00 06 00 0D 00
06 00 08 00 01 00 07 00 0B 00 04 00 00 00 00 00
03 00 07 00 0E 00 0D 00 00 00 02 00 02 00 02 00
02 00 01 00 01 00 02 00 02 00 00 00 00 00 00 00
00 00 00 00 00 00 04 00 01 00 00 00 00 00 82 00
2B 00 00 00 0C 00 00 00 00 00 02 00 02 0D 00 00
00 00 00 00 02 03 C0 00 04 24 12 00 F7 64 86 00
D2 39 81 00 FA 12 02 01 01 00 00 00 2C 01 2B 00
FF 7F 00 00 00 DB E2 33 F7 64 86 00 05 00 5D 3E
FF 7F 00 00 6A 01 06 00 00 00 18 02 FC 4C 03 00
00 00 43 48 78 72 03 18 0F 20 04 2C C1 02 4F 22
00 00 00 00 6F FC 72 D0 C5 08 2E 57 46 1D 52 2C
34 6B 31 7A 33 61 68 71 37 61 31 70 68 52 45 33
C5 02 5F 40 13 42 45 52 00 00 00 00 78 B8 43 F8
80 2E 52 4F 78 72 07 02 0E 12 0E 49 46 49 58 45
44 20 54 41 42 4C 45 22 46 55 4C 4C 53 59 53 4E
58 24 4B 5A 53 50 52 32 58 24 4B 5A 53 50 52 40
53 45 4C 24 37 33 5B 4E 54 41 42 4C 45 20 28 46
01 22 00 00 8C 04 00 00 66 0F 00 00 00 81 62 5E
49 58 45 44 29 22 2E 22 C1 4A 41 52 C1 04 22 5B
C1 05 4D 42 C1 1C 2C 32 80 5D 2C 20 .  .  .  .
```

→ UNDO change (5.1 - DRP - delete)
→ REDO change (11.2 - IRP - insert)

→ Object ID 0x00041104
→ Object DATA ID 0x00041104

→ 45 data parts ((5C / 2) - 1)

Change data

```
5C 00 10 00 36 00 07 00 02 00 08 00 06 00 0D 00
06 00 08 00 01 00 07 00 0B 00 04 00 00 00 00 00
03 00 07 00 0E 00 0D 00 00 00 02 00 02 00 02 00
02 00 01 00 01 00 02 00 02 00 00 00 00 00 00 00
00 00 00 00 00 00 04 00 01 00 00 00 00 00 82 00
2B 00 00 00 0C 00 00 00 00 00 02 00 02 0D 00 00
00 00 00 00 02 03 C0 00 04 24 12 00 F7 64 86 00
D2 39 81 00 FA 12 02 01 01 00 00 00 2C 01 2B 00
FF 7F 00 00 00 DB E2 33 F7 64 86 00 05 00 5D 3E
FF 7F 00 00 6A 01 06 00 00 00 18 02 FC 4C 03 00
00 00 43 48 78 72 03 18 0F 20 04 2C C1 02 4F 22
80 00 00 00 6F FC 72 D0 C5 08 2E 57 46 1D 52 2C
34 6B 31 7A 33 61 68 71 37 61 31 70 68 52 45 33
C5 02 5F 40 13 42 45 52 00 00 00 00 78 B8 43 F8
80 2E 52 4F 78 72 07 02 0E 12 0E 49 46 49 58 45
44 20 54 41 42 4C 45 22 46 55 4C 4C 53 59 53 4E
58 24 4B 5A 53 50 52 32 58 24 4B 5A 53 50 52 40
53 45 4C 24 37 33 5B 4E 54 41 42 4C 45 20 28 46
49 58 45 44 29 22 2E 22 C1 4A 41 52 C1 04 22 5B
C1 05 4D 42 C1 1C 2C 32 80 5D 2C 20 80 55 22 2E
C1 02 50 41 C1 1B 32 22 C3 27 04 5F 80 45 52 2C
28 22 49 4E 53 54 5F 49 44 22 3D 55 53 45 52 45
4E 56 28 27 49 4E 53 54 41 4E 43 45 27 29 20 41
4E 44 20 28 28 2D 22 4B 5A 53 50 52 50 52 56 22
29 3D 28 2D 32 35 38 29 20 4F 52 20 28 2D 22 4B
5A 53 50 52 50 52 56 22 29 3D 28 2D 32 35 39 29
20 4F 52 20 28 2D 22 4B 5A 53 50 52 50 52 56 22
29 3D 28 2D 32 36 30 29 20 4F 52 20 28 2D 22 4B
5A 53 50 52 50 52 56 22 29 3D 28 2D 32 36 31 29
29 29 45 BA 22 49 4E 53 54 5F 49 44 22 5B 4E 55
4D 42 45 52 2C 32 32 5D 2C 20 22 4B 5A 53 50 52
50 52 56 22 5B 4E 55 4D 42 45 52 2C ...
```

→ 24.3.2014 14:31:03 → TABLE (FIXED)
→ 1 → 65
→ ?? (probably RAW) → 3
→ 745866928 → 4
→ 4k1z3ahq7a1ph → 27
→ 194631865 → NULL
→ ?? (probably RAW) → NULL
→ NULL → 1
→ 2.7.2014 13:17:13 → 26
→ FIXED TABLE → 380394
→ FULL → NULL
→ SYS
→ X$KZSPR
→ X$KZSPR@SEL$73

→ ("INST_ID"=USERENV('INSTANCE') AND ((-"KZSPRPRV")=(-258) OR (-"KZSPRPRV")=(-259) OR (-"KZSPRPRV")=(-260) OR (-"KZSPRPRV")=(-261)))

→ "INST_ID"[NUMBER,22], "KZSPRPRV"[NUMBER,22]

What to look for?

Data operations:

- 11.2 - INSERT
- 11.3 - DELETE
- 11.5 - UPDATE
- 11.11 - MULTI-INSERT
- 11.12 - MULTI-DELETE
- 11.6 - OVERWRITE
- 5.1 - UNDO
- 19.1 - DIRECT LOADER

Transaction operations:

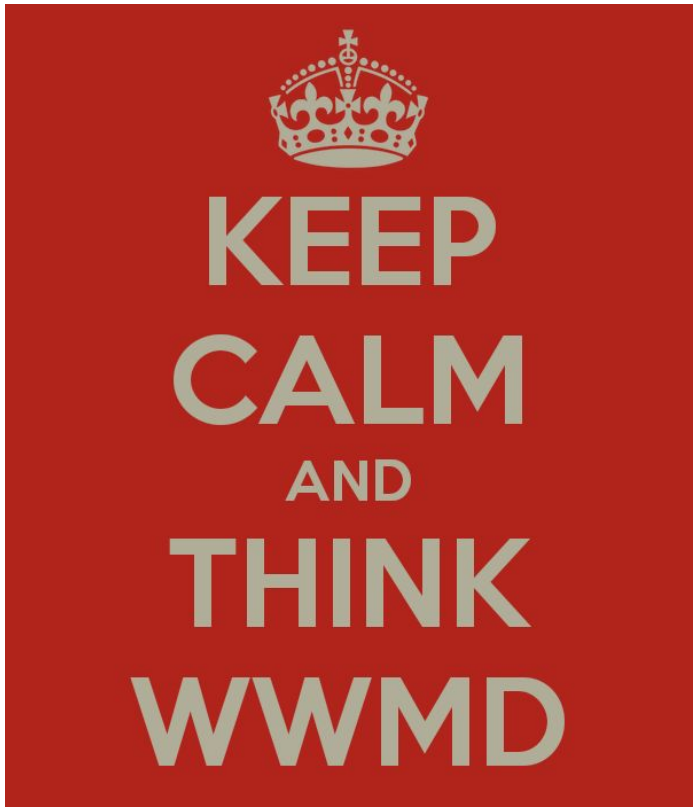
- 5.2 - HEADER
- 5.4 - COMMIT/ROLLBACK
- 5.6 - ROLLBACK TO SAVEPOINT
- 5.19 - SESSION BEGIN
- 5.20 - SESSION SWITCH

What about the details?

some light eading

- **David Litchfield - Dissecting the Redo Logs**
  http://www.davidlitchfield.com/oracle_forensics_part_1._dissecting_the_redo_logs.pdf
- **Redo Internals - Julian Dyke**
  http://www.juliandyke.com/Presentations/RedoInternals.ppt
- **Zizzy**
  http://sourceforge.net/projects/zizzy/

Time for tooling ...



- The Good
  - A good HEX editor
- The bad
  - Spreadsheet app (Calc, Excel)
- ~~The ugly~~
  - Pen & paper (and some Xanax)

DO NOT GO THERE!

… or better yet:

MAKE YOUR OWN TOOLS!!!

Security observations

- Data **CAN** be read from redo logs

  you have to guess data types if you don't know the table columns

- Data **CAN** be modified in redo logs

  you are limited to current data lengths or you have to shift records

  you must update CRC

- DDL is in clear text

long story short … just keep the log files safe

```
-rw-r----- 1 oracle dba 10485100 jan 01 00:00 1_23456_789012345.arch
```

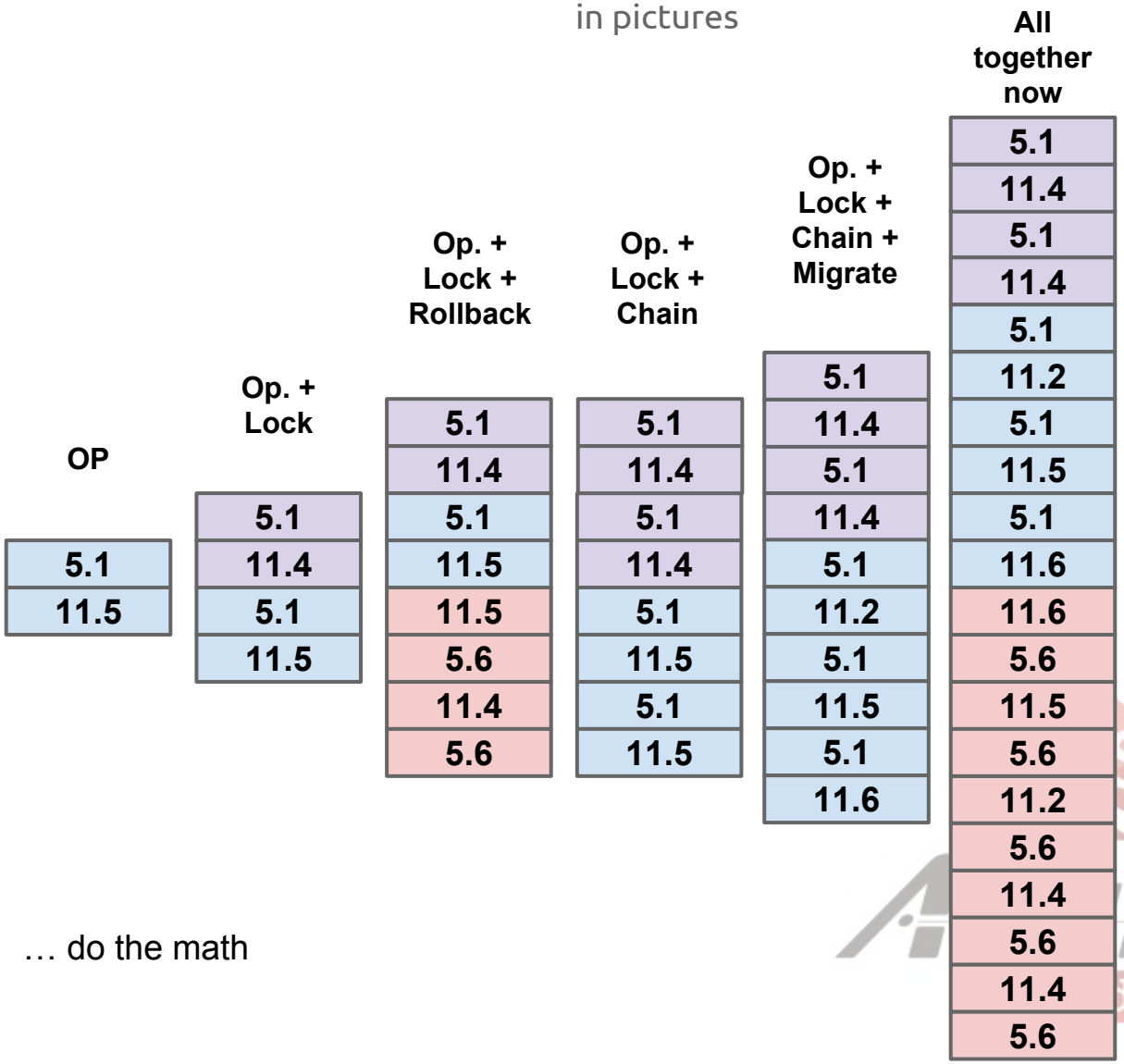
Optimisation observations

- Try to avoid row chaining
  - multiplies data change count by mod(cols,255) **OR MORE**
  - if you use tables with >255 columns **TRY HARDER**

- Try to avoid ROLLBACK TO SAVEPOINT
  - do not use it for flow control logic

Optimisation observations

in pictures



… do the math

Questions?!

?

Jure Kajzer
Abakus PLUS d.o.o.

jure.kajzer@abakus.si

AS
Abakus
As na disku.