SUPERCHARGING BIG PRODUCTION SYSTEMS ON FIREBIRD: TRANSACTIONS, GARBAGE, MAINTENANCE

Dmitry Kuzmenko, IBSurgeon

What is big?

- 10 users, 1gb database?
- 100 users, 1gb database?
- 5 users, 100gb database?

- **Any system, that have performance problems**

Steps to tune performance

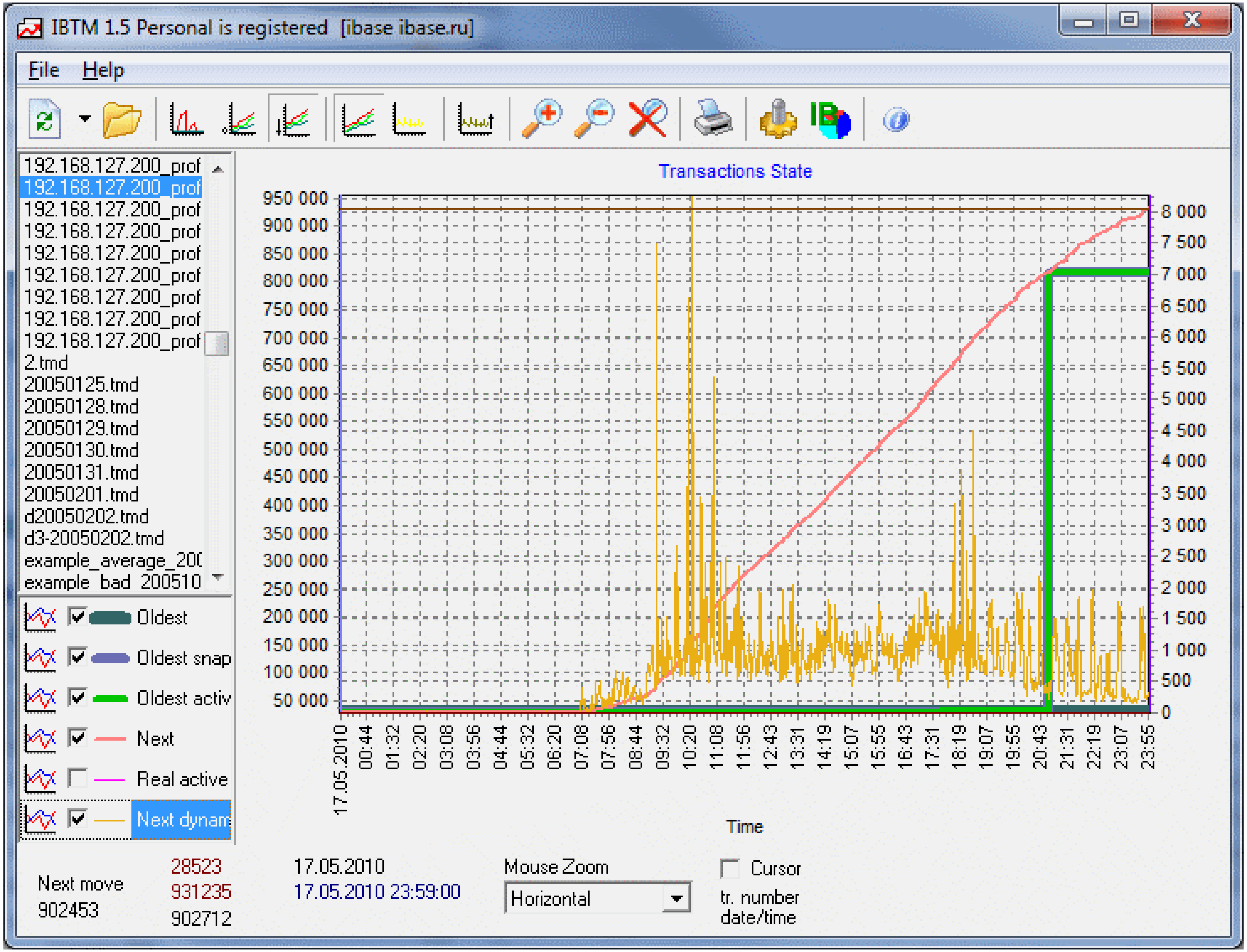
1. Hardware
    1. Processor, Memory, Disks
    2. Virtual machines
2. Transactions
3. Versioning
4. Queries

- Everything need to be able logged and monitored

Transactions

- What is happening in the system
- What about transaction control in applications (good/bad)
- Are there unnecessary active transactions

IBTM: Transactions monitoring



192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
192.168.127.200_prof
2.tmd
20050125.tmd
20050128.tmd
20050129.tmd
20050130.tmd
20050131.tmd
20050201.tmd
d20050202.tmd
d3-20050202.tmd
example_average_200
example_bad_200510

☑ Oldest
☑ Oldest snap
☑ Oldest activ
☑ Next
□ Real active
☑ Next dynam

Next move
902453

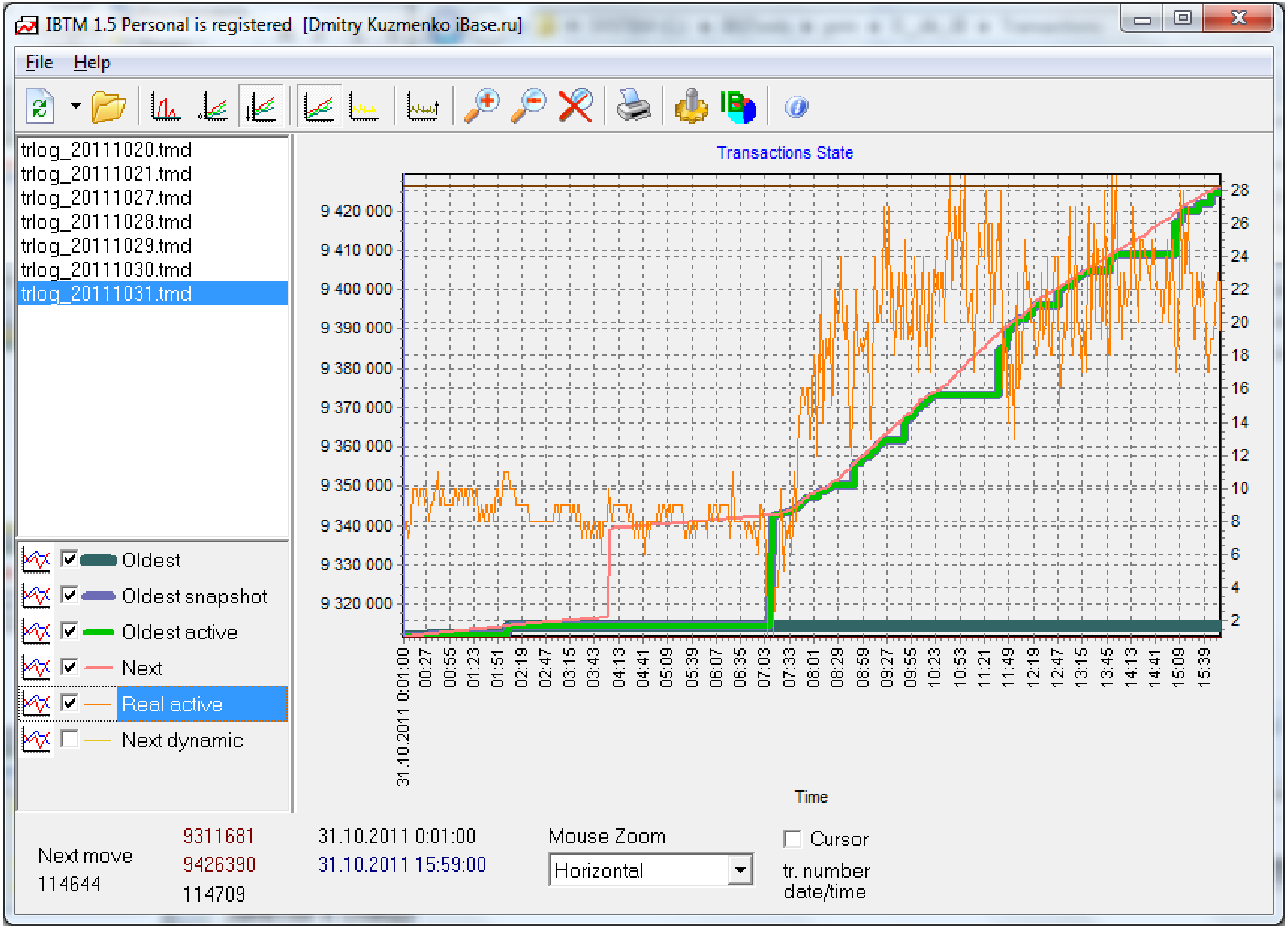
28523
931235
902712

17.05.2010
17.05.2010 23:59:00

Mouse Zoom
Horizontal

□ Cursor
tr. number
date/time

Active transactions



trlog_20111020.tmd
trlog_20111021.tmd
trlog_20111027.tmd
trlog_20111028.tmd
trlog_20111029.tmd
trlog_20111030.tmd
trlog_20111031.tmd

☑ Oldest
☑ Oldest snapshot
☑ Oldest active
☑ Next
☑ Real active
□ Next dynamic

Next move
114644

9311681
9426390
114709

31.10.2011 0:01:00
31.10.2011 15:59:00

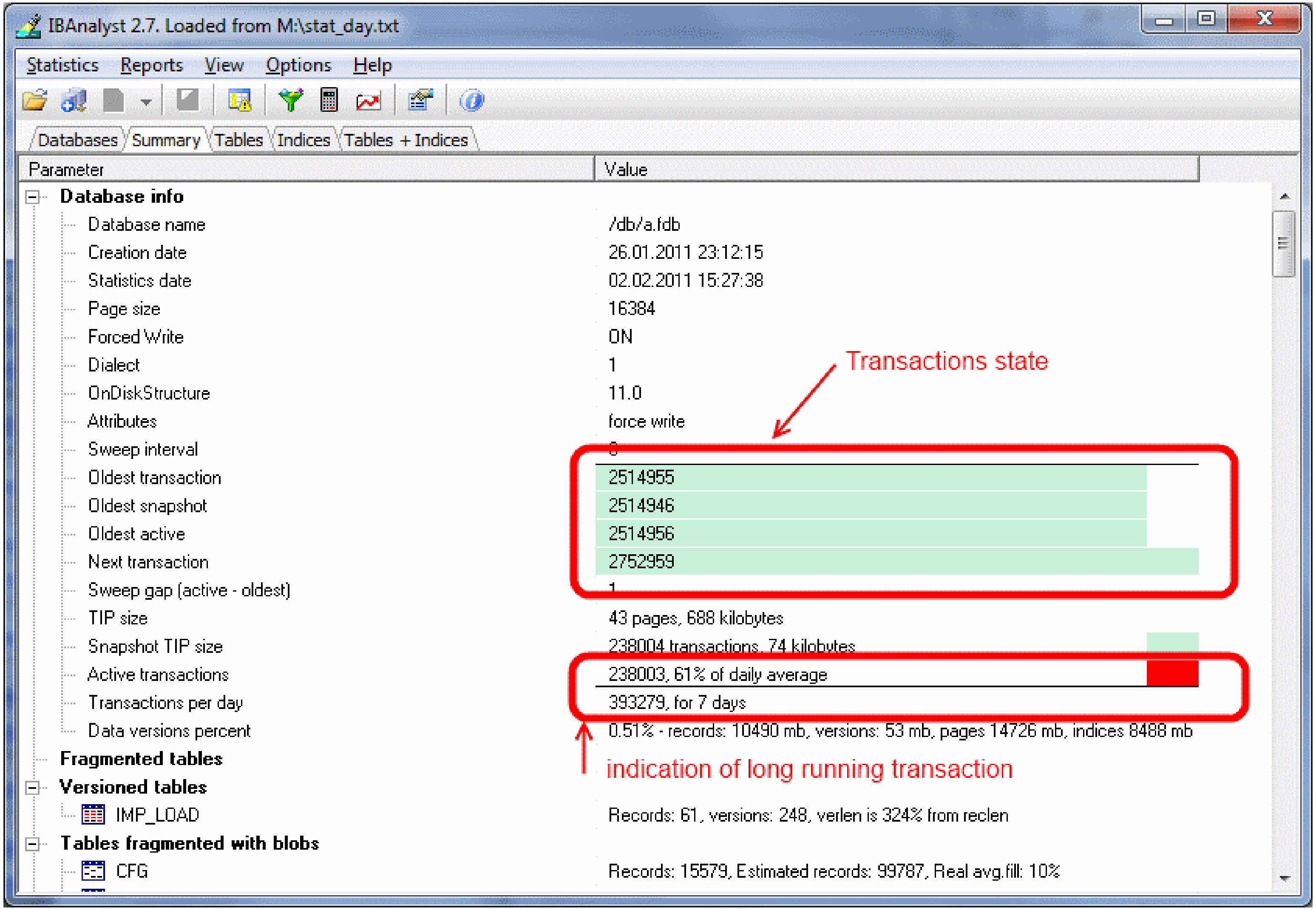
Mouse Zoom
Horizontal

□ Cursor
tr. number
date/time

Versioning

- Do we need to set sweep interval to 0?
- If yes, when it’s better to run gfix –sweep ?
- Where versions are, and how much
- What is garbage?
- What is the cause of versions?

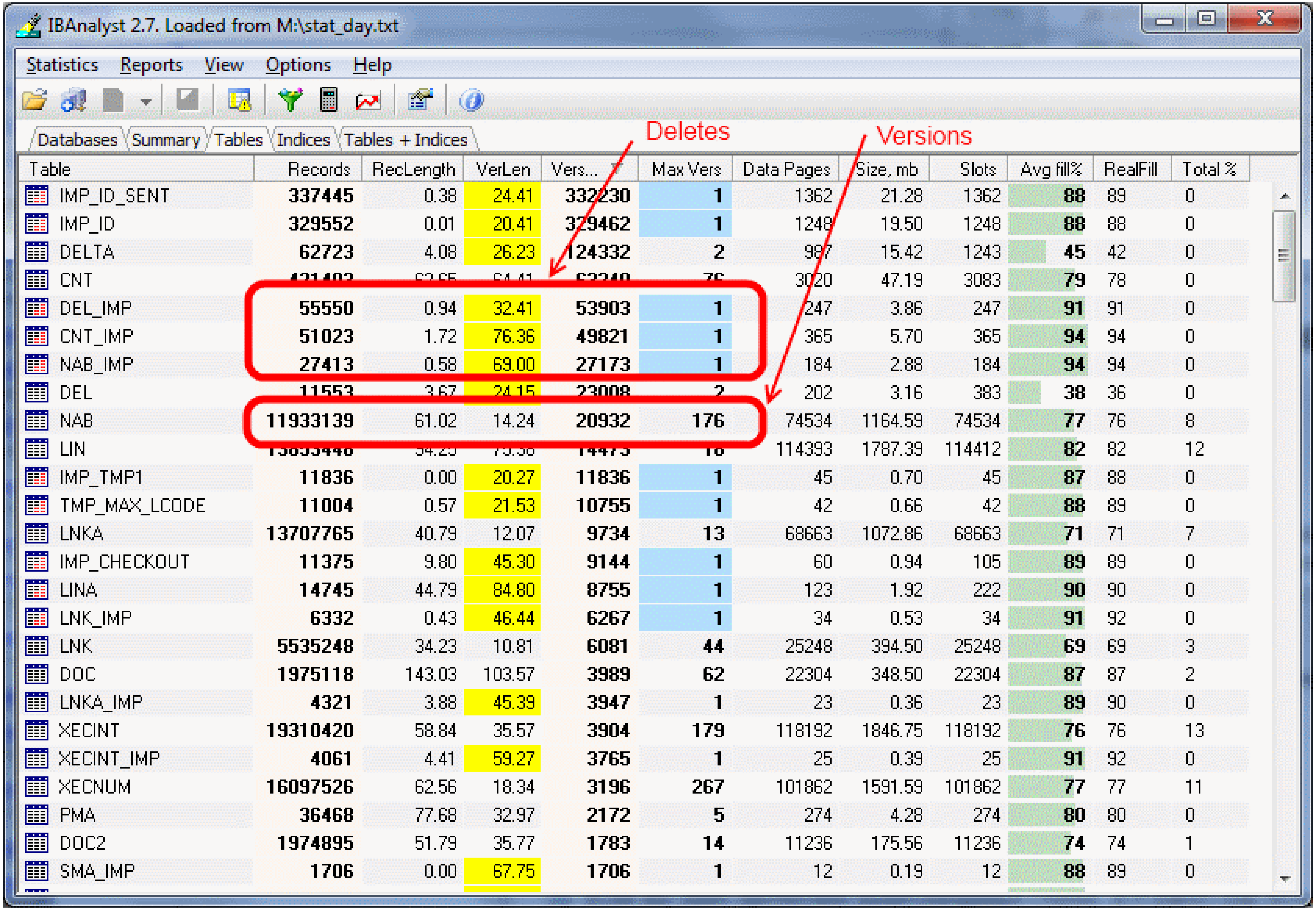
IBAnalyst



Parameter	Value
Database info	
Database name	/db/a.fdb
Creation date	26.01.2011 23:12:15
Statistics date	02.02.2011 15:27:38
Page size	16384
Forced Write	ON
Dialect	1
OnDiskStructure	11.0
Attributes	force write
Sweep interval	0
Oldest transaction	2514955
Oldest snapshot	2514946
Oldest active	2514956
Next transaction	2752959
Sweep gap (active - oldest)	1
TIP size	43 pages, 688 kilobytes
Snapshot TIP size	238004 transactions, 74 kilobytes
Active transactions	238003, 61% of daily average
Transactions per day	393279, for 7 days
Data versions percent	0.51% - records: 10490 mb, versions: 53 mb, pages 14726 mb, indices 8488 mb
Fragmented tables	
Versioned tables	
IMP_LOAD	Records: 61, versions: 248, verlen is 324% from reclen
Tables fragmented with blobs	
CFG	Records: 15579, Estimated records: 99787, Real avg.fill: 10%

Transactions state

indication of long running transaction

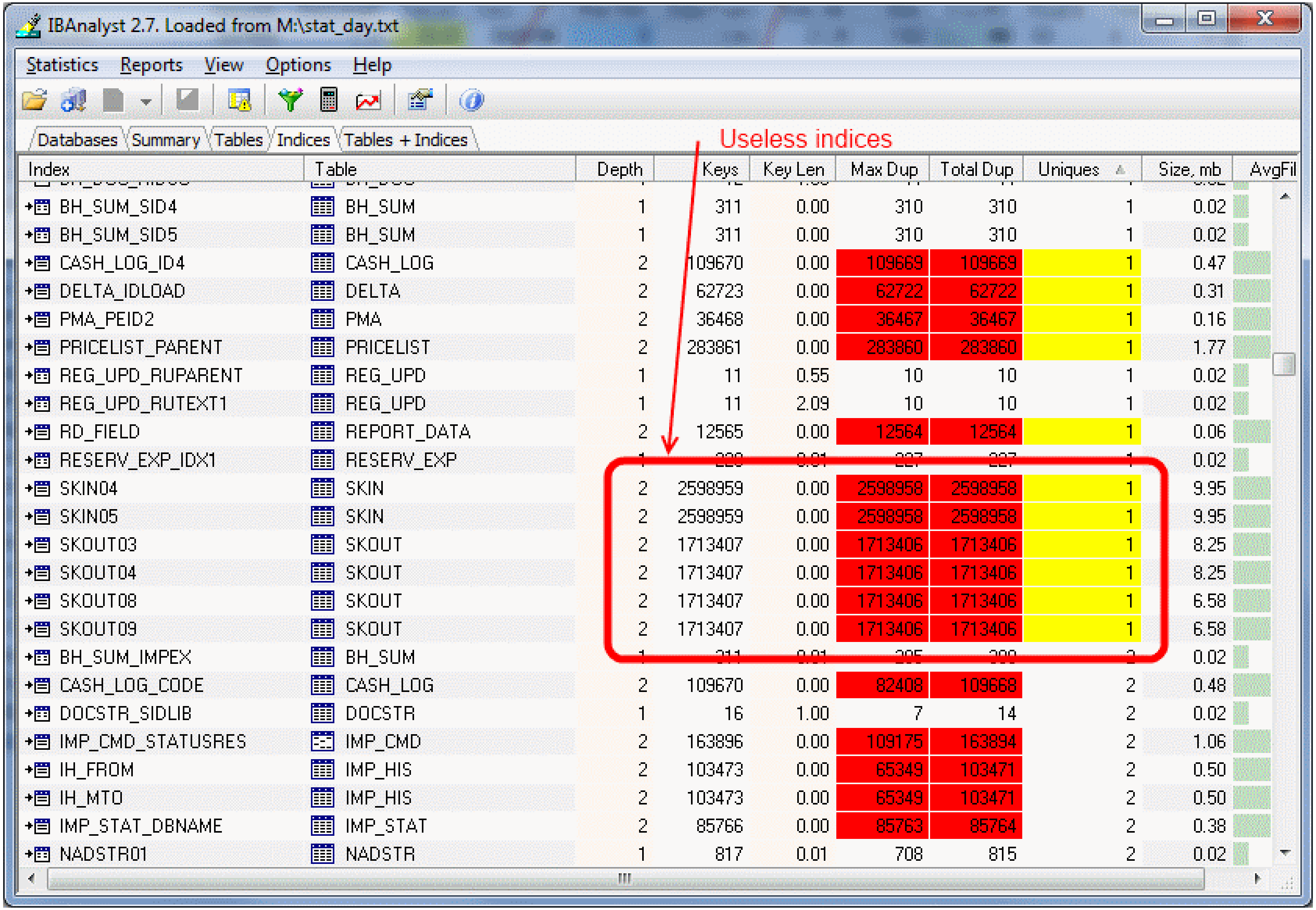


IBAnalyst 2.7. Loaded from M:\stat_day.txt
Statistics Reports View Options Help
Databases
Summary
Tables
Indices
Tables + Indices

Deletes

Versions

Table	Records	RecLength	VerLen	Vers...	Max Vers	Data Pages	Size, mb	Slots	Avg fill%	RealFill	Total %
IMP_ID_SENT	337445	0.38	24.41	332230	1	1362	21.28	1362	88	89	0
IMP_ID	329552	0.01	20.41	329462	1	1248	19.50	1248	88	88	0
DELTA	62723	4.08	26.23	124332	2	987	15.42	1243	45	42	0
CNT	421403	62.55	64.41	63240	76	3020	47.19	3083	79	78	0
DEL_IMP	55550	0.94	32.41	53903	1	247	3.86	247	91	91	0
CNT_IMP	51023	1.72	76.36	49821	1	365	5.70	365	94	94	0
NAB_IMP	27413	0.58	69.00	27173	1	184	2.88	184	94	94	0
DEL	11553	3.67	24.15	23008	2	202	3.16	383	38	36	0
NAB	11933139	61.02	14.24	20932	176	74534	1164.59	74534	77	76	8
LIN	13653448	34.25	75.38	14475	16	114393	1787.39	114412	82	82	12
IMP_TMP1	11836	0.00	20.27	11836	1	45	0.70	45	87	88	0
TMP_MAX_LCODE	11004	0.57	21.53	10755	1	42	0.66	42	88	89	0
LNKA	13707765	40.79	12.07	9734	13	68663	1072.86	68663	71	71	7
IMP_CHECKOUT	11375	9.80	45.30	9144	1	60	0.94	105	89	89	0
LINA	14745	44.79	84.80	8755	1	123	1.92	222	90	90	0
LNK_IMP	6332	0.43	46.44	6267	1	34	0.53	34	91	92	0
LNK	5535248	34.23	10.81	6081	44	25248	394.50	25248	69	69	3
DOC	1975118	143.03	103.57	3989	62	22304	348.50	22304	87	87	2
LNKA_IMP	4321	3.88	45.39	3947	1	23	0.36	23	89	90	0
XECINT	19310420	58.84	35.57	3904	179	118192	1846.75	118192	76	76	13
XECINT_IMP	4061	4.41	59.27	3765	1	25	0.39	25	91	92	0
XECNUM	16097526	62.56	18.34	3196	267	101862	1591.59	101862	77	77	11
PMA	36468	77.68	32.97	2172	5	274	4.28	274	80	80	0
DOC2	1974895	51.79	35.77	1783	14	11236	175.56	11236	74	74	1
SMA_IMP	1706	0.00	67.75	1706	1	12	0.19	12	88	89	0



Useless indices

Index	Table	Depth	Keys	Key Len	Max Dup	Total Dup	Uniques	Size, mb	AvgFil
BH_SUM_SID4	BH_SUM	1	311	0.00	310	310	1	0.02	🟩
BH_SUM_SID5	BH_SUM	1	311	0.00	310	310	1	0.02	🟩
CASH_LOG_ID4	CASH_LOG	2	109670	0.00	109669	109669	1	0.47	🟩
DELTA_IDLOAD	DELTA	2	62723	0.00	62722	62722	1	0.31	🟩
PMA_PEID2	PMA	2	36468	0.00	36467	36467	1	0.16	🟩
PRICELIST_PARENT	PRICELIST	2	283861	0.00	283860	283860	1	1.77	🟩
REG_UPD_RUPARENT	REG_UPD	1	11	0.55	10	10	1	0.02	🟩
REG_UPD_RUTEXT1	REG_UPD	1	11	2.09	10	10	1	0.02	🟩
RD_FIELD	REPORT_DATA	2	12565	0.00	12564	12564	1	0.06	🟩
RESERV_EXP_IDX1	RESERV_EXP	1	228	0.01	227	227	1	0.02	🟩
SKIN04	SKIN	2	2598959	0.00	2598958	2598958	1	9.95	🟩
SKIN05	SKIN	2	2598959	0.00	2598958	2598958	1	9.95	🟩
SKOUT03	SKOUT	2	1713407	0.00	1713406	1713406	1	8.25	🟩
SKOUT04	SKOUT	2	1713407	0.00	1713406	1713406	1	8.25	🟩
SKOUT08	SKOUT	2	1713407	0.00	1713406	1713406	1	6.58	🟩
SKOUT09	SKOUT	2	1713407	0.00	1713406	1713406	1	6.58	🟩
BH_SUM_IMPEX	BH_SUM	1	311	0.01	305	309	2	0.02	🟩
CASH_LOG_CODE	CASH_LOG	2	109670	0.00	82408	109668	2	0.48	🟩
DOCSTR_SIDLIB	DOCSTR	1	16	1.00	7	14	2	0.02	🟩
IMP_CMD_STATUSRES	IMP_CMD	2	163896	0.00	109175	163894	2	1.06	🟩
IH_FROM	IMP_HIS	2	103473	0.00	65349	103471	2	0.50	🟩
IH_MTO	IMP_HIS	2	103473	0.00	65349	103471	2	0.50	🟩
IMP_STAT_DBNAME	IMP_STAT	2	85766	0.00	85763	85764	2	0.38	🟩
NADSTR01	NADSTR	1	817	0.01	708	815	2	0.02	🟩

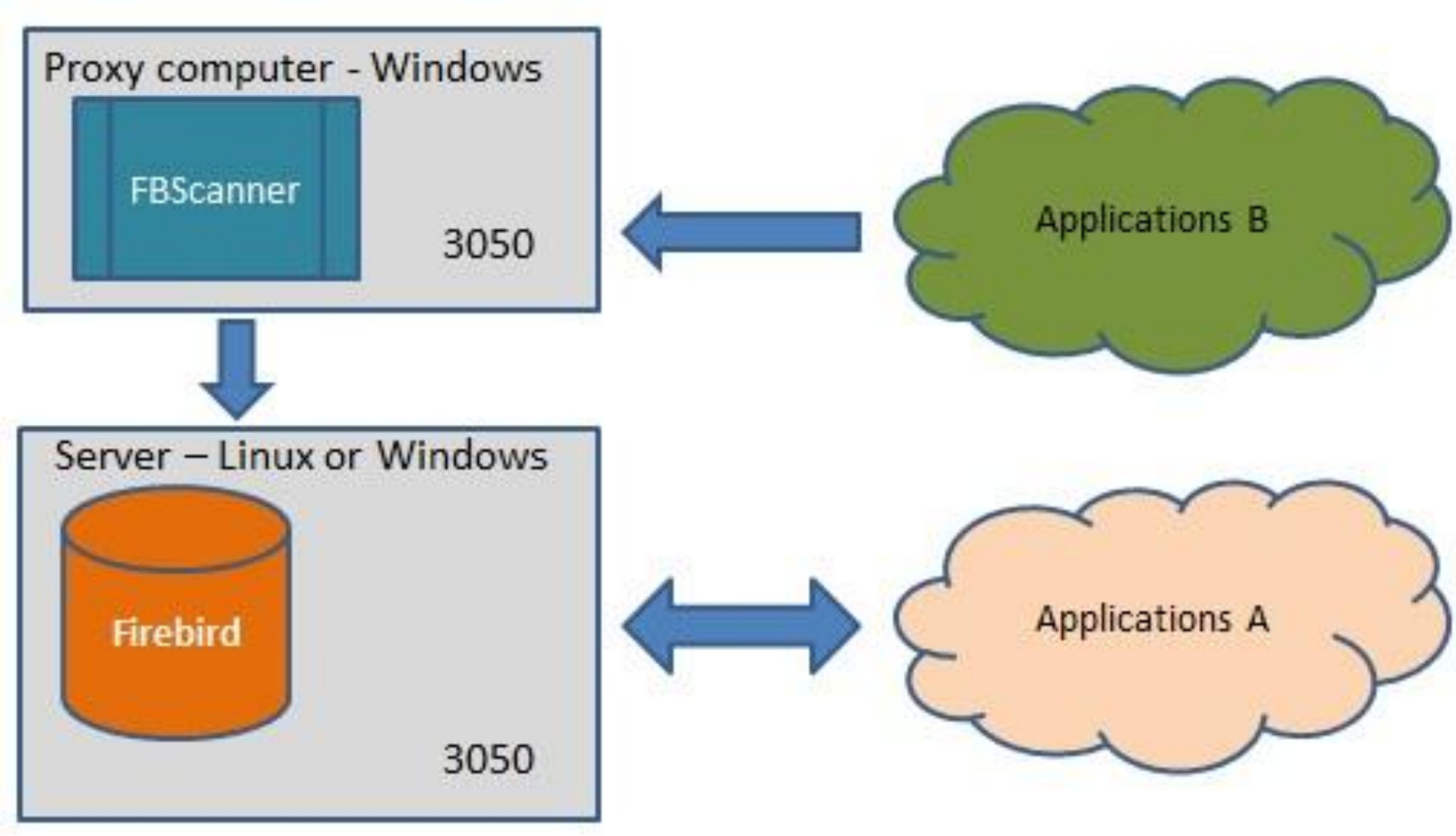
What business logic causes versions grow?

- Understandable via special UDFs
  - Udf writes some parameters to some file
  - Udfs are called in INSERT/UPDATE/DELETE triggers on tables, that we want to monitor.
  - UDFs can be called from procedures to understand begin/end of logic block
  - Intensity can be analyzed by file size
  - Timestamps and blocks can help to understand logic
- Trace API

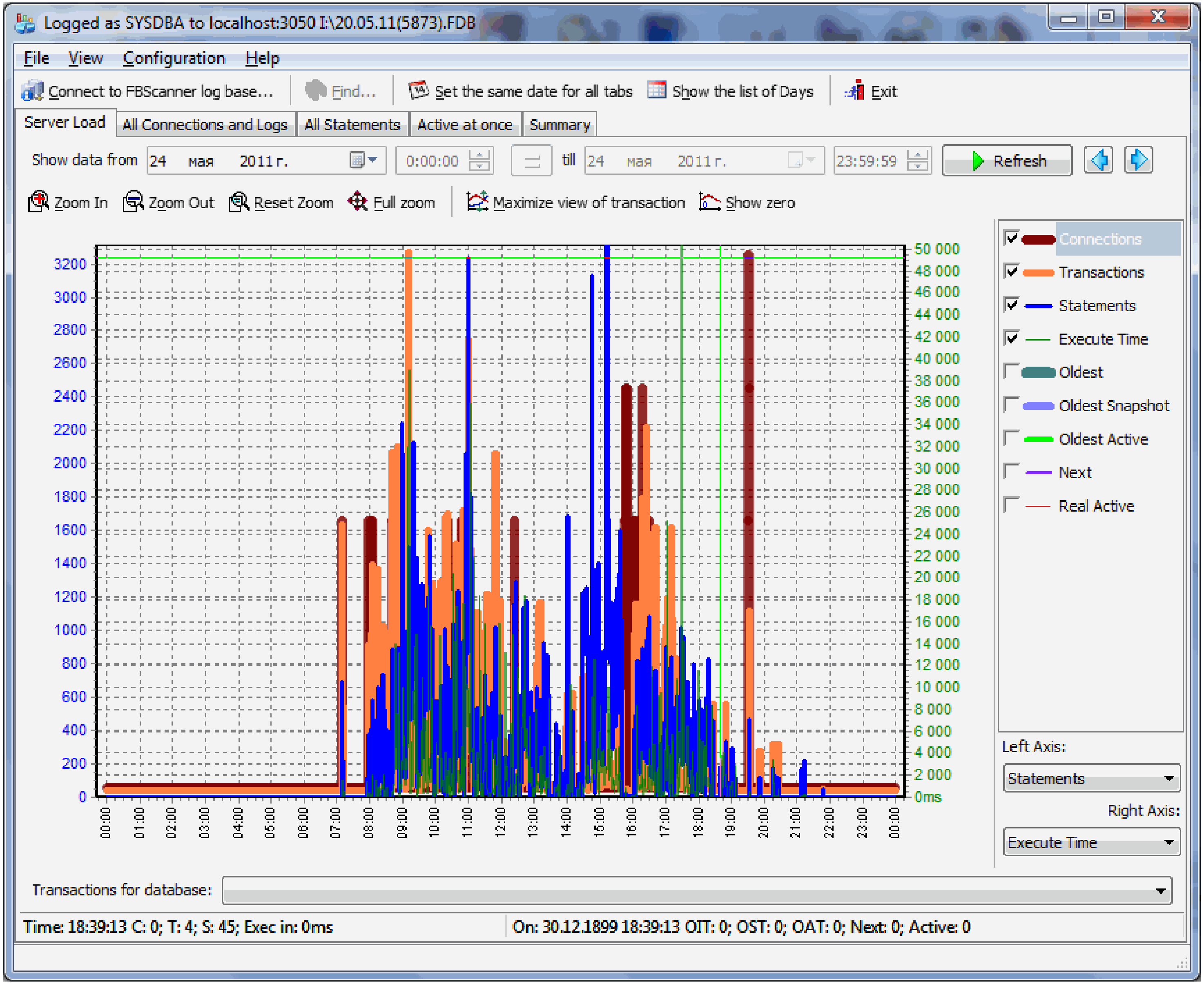
Queries

- How many queries my application produce?
- Does all queries correct and even necessary (obsolete/wrong functionality)?
- Which queries take too much time?
- How query execution time degrades (if yes)?

FBScanner


Proxy computer - Windows
FBScanner
3050
Applications B
Server – Linux or Windows
Firebird
3050
Applications A

1 day, 84 computers, up to 1350 transactions per minute, up to 3200 statements per minute, longest query - 50 seconds



Logged as SYSDBA to localhost:3050 I:\20.05.11(5873).FDB

Show data from 24 мая 2011 г. 0:00:00 till 24 мая 2011 г. 23:59:59

☑ Connections
☑ Transactions
☑ Statements
☑ Execute Time
□ Oldest
□ Oldest Snapshot
□ Oldest Active
□ Next
□ Real Active

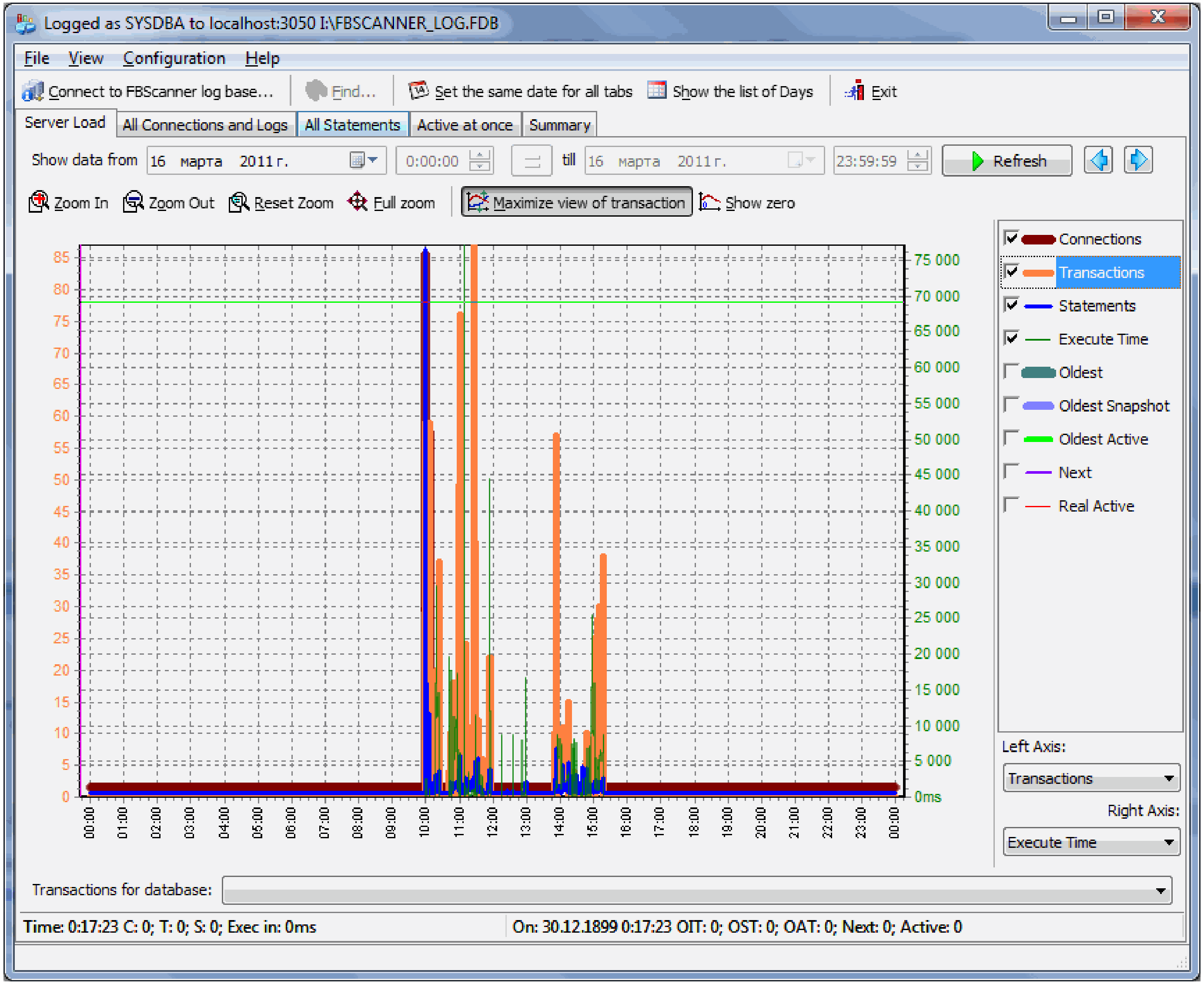
Left Axis: Statements

Right Axis: Execute Time

Transactions for database:

Time: 18:39:13 C: 0; T: 4; S: 45; Exec in: 0ms On: 30.12.1899 18:39:13 OIT: 0; OST: 0; OAT: 0; Next: 0; Active: 0

1 application/user monitored
up to 85 transactions per minute (?), up to 4200 sql-operators per minute (???)



Logged as SYSDBA to localhost:3050 I:\FBSCANNER_LOG.FDB

Show data from 16 марта 2011 г. 0:00:00 till 16 марта 2011 г. 23:59:59

Left Axis: Transactions

Right Axis: Execute Time

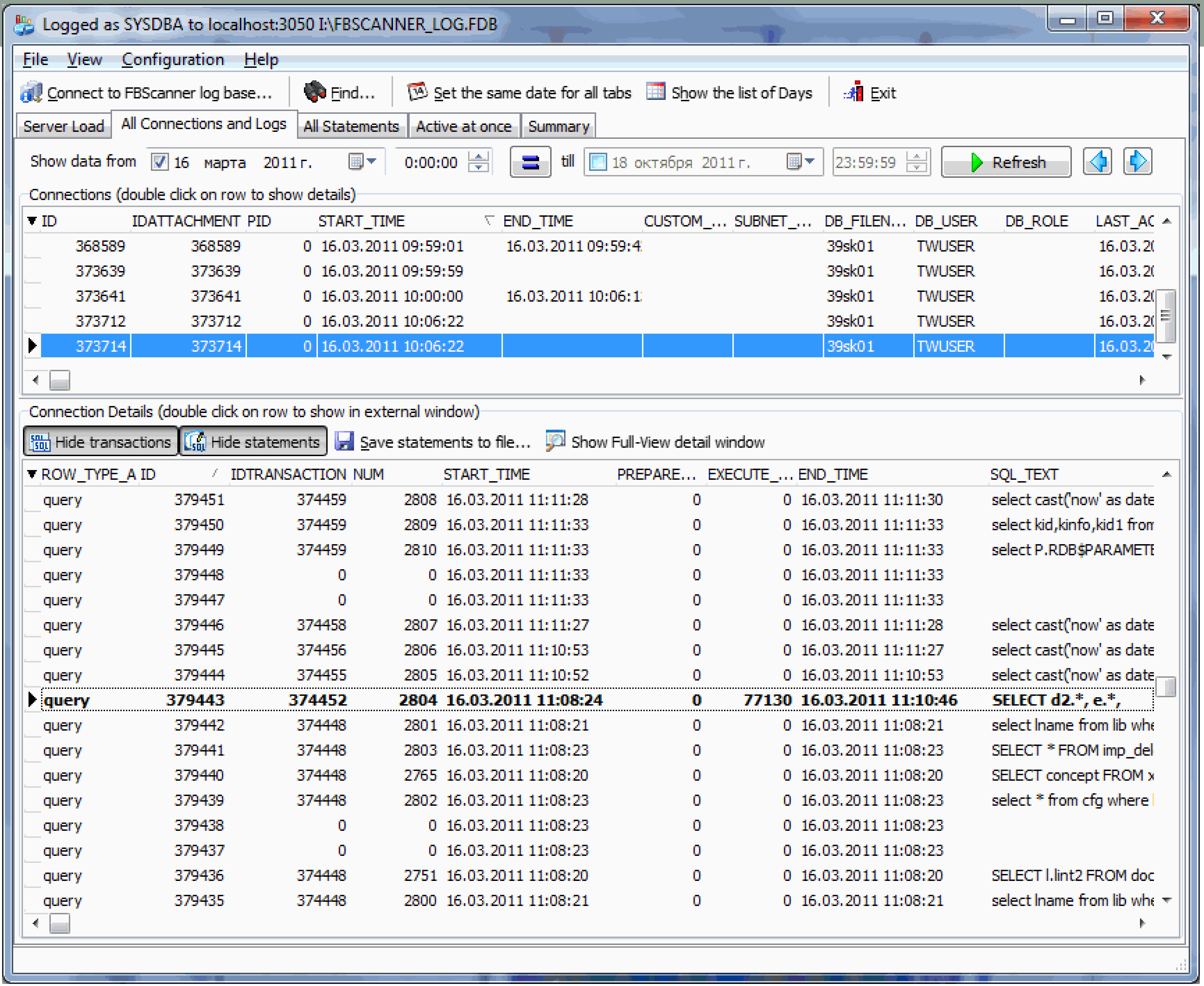
Transactions for database:

Time: 0:17:23 C: 0; T: 0; S: 0; Exec in: 0ms

On: 30.12.1899 0:17:23 OIT: 0; OST: 0; OAT: 0; Next: 0; Active: 0

What observer can discover?

- Users use not only one instance of application
- Application runs unnecessary queries
- Application use wrong transaction control
- Users use application not the way it was designed by developers


Logged as SYSDBA to localhost:3050 I:\FBSCANNER_LOG.FDB
File
View
Configuration
Help
Connect to FBScanner log base...
Find...
Set the same date for all tabs
Show the list of Days
Exit
Server Load
All Connections and Logs
All Statements
Active at once
Summary
Refresh
Hide transactions
Hide statements
Save statements to file...
Show Full-View detail window

Show data from ☑ 16 марта 2011 г. 0:00:00 till □ 18 октября 2011 г. 23:59:59

Connections (double click on row to show details)

ID	IDATTACHMENT	PID	START_TIME	END_TIME	CUSTOM_...	SUBNET_...	DB_FILEN...	DB_USER	DB_ROLE	LAST_AC
368589	368589	0	16.03.2011 09:59:01	16.03.2011 09:59:4			39sk01	TWUSER		16.03.20
373639	373639	0	16.03.2011 09:59:59				39sk01	TWUSER		16.03.20
373641	373641	0	16.03.2011 10:00:00	16.03.2011 10:06:1			39sk01	TWUSER		16.03.20
373712	373712	0	16.03.2011 10:06:22				39sk01	TWUSER		16.03.20
373714	373714	0	16.03.2011 10:06:22				39sk01	TWUSER		16.03.20

Connection Details (double click on row to show in external window)

ROW_TYPE_A	ID	IDTRANSACTION	NUM	START_TIME	PREPARE...	EXECUTE_...	END_TIME	SQL_TEXT
query	379451	374459	2808	16.03.2011 11:11:28	0	0	16.03.2011 11:11:30	select cast('now' as date
query	379450	374459	2809	16.03.2011 11:11:33	0	0	16.03.2011 11:11:33	select kid,kinfo,kid1 from
query	379449	374459	2810	16.03.2011 11:11:33	0	0	16.03.2011 11:11:33	select P.RDB$PARAMETE
query	379448	0	0	16.03.2011 11:11:33	0	0	16.03.2011 11:11:33	
query	379447	0	0	16.03.2011 11:11:33	0	0	16.03.2011 11:11:33	
query	379446	374458	2807	16.03.2011 11:11:27	0	0	16.03.2011 11:11:28	select cast('now' as date
query	379445	374456	2806	16.03.2011 11:10:53	0	0	16.03.2011 11:11:27	select cast('now' as date
query	379444	374455	2805	16.03.2011 11:10:52	0	0	16.03.2011 11:10:53	select cast('now' as date
query	**379443**	**374452**	**2804**	**16.03.2011 11:08:24**	**0**	**77130**	**16.03.2011 11:10:46**	**SELECT d2.*, e.*,**
query	379442	374448	2801	16.03.2011 11:08:21	0	0	16.03.2011 11:08:21	select lname from lib whe
query	379441	374448	2803	16.03.2011 11:08:23	0	0	16.03.2011 11:08:23	SELECT * FROM imp_del
query	379440	374448	2765	16.03.2011 11:08:20	0	0	16.03.2011 11:08:20	SELECT concept FROM x
query	379439	374448	2802	16.03.2011 11:08:23	0	0	16.03.2011 11:08:23	select * from cfg where
query	379438	0	0	16.03.2011 11:08:23	0	0	16.03.2011 11:08:23	
query	379437	0	0	16.03.2011 11:08:23	0	0	16.03.2011 11:08:23	
query	379436	374448	2751	16.03.2011 11:08:20	0	0	16.03.2011 11:08:20	SELECT l.lint2 FROM doc
query	379435	374448	2800	16.03.2011 11:08:21	0	0	16.03.2011 11:08:21	select lname from lib whe

What to do next

- Ajust (fix) transaction management in applications
  - Remove unnecessary write transactions
  - Fix duration of transactions
  - Legacy applications, lazy users
- Garbage
  - Drop unnecessary indices
  - If source is available - fix business logic
  - Closed source and legacy applications – require special maintenance schedule
- Queries
  - Fix plans or change queries
  - create/drop indices
  - Try to use stored aggregates

Summary

- IBSurgeon’s supercharge service is based on IBTM, IBAnalyst and FBScanner, to identify and fix problems with transactions, garbage and maintenance
- Tools are available to be adopted by any experienced Firebird professional

Special offer:

–25% for conference attendees till December 31

support@ib-aid.com

www.ib-aid.com