PLATINUM

IBSurgeon

From Firebird 1.5 to 2.5

How to migrate 75Gb database, with 564 tables, 5000+ stored procedures, 813 triggers, which is working 24x7, with ~400 users in less than 4 months

About IBSurgeon

PLATINUM

IBSurgeon

- Tools and consulting
- Platinum Sponsor of Firebird
- Founded in 2002: 9 years of Firebird recoveries and consulting
- Based in Moscow, Russia

www.ib-aid.com www.ibsurgeon.com

About Profitmed

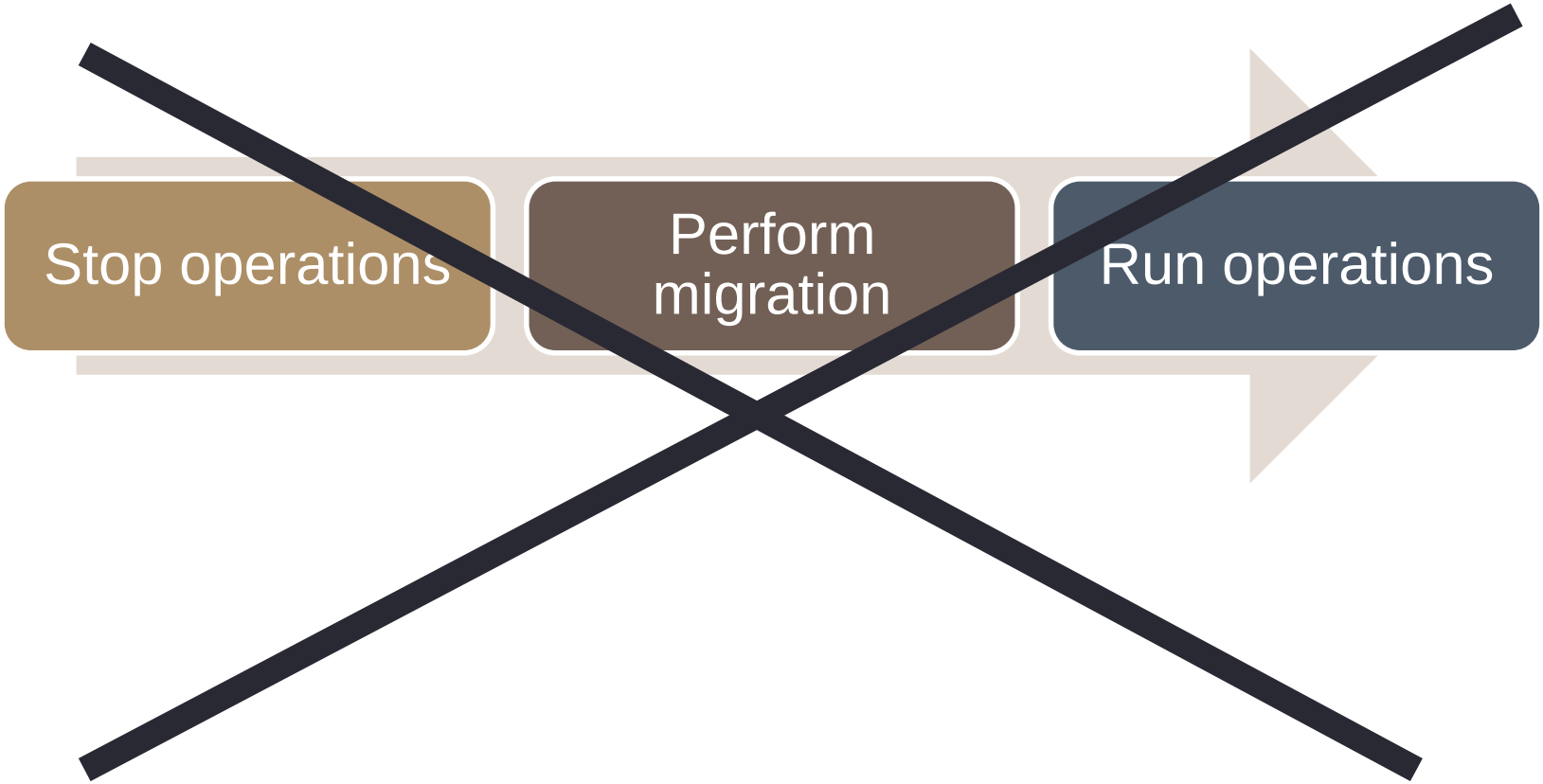
- http://www.profitmed.net/
- Pharmaceutical distributor Russia
- 24x7 work mode for 2 warehouses (16000 sqm + 7000 sqm)
- 12x7 work mode for office
- **400** peak users
- **75 Gb** database size
- 64 Gb RAM, RAID 10 (Dell Storage), Xeon 4x6 cores

Why to migrate?

- From Firebird 1.5
  - Old (2004), no professional support
  - Bad memory management for huge queries
  - Slow garbage collection (1.5 hour+)
  - Slow backup (2 hours)
  - Firebird Classic 1.5 [actually] does not use more than 6 CPU cores

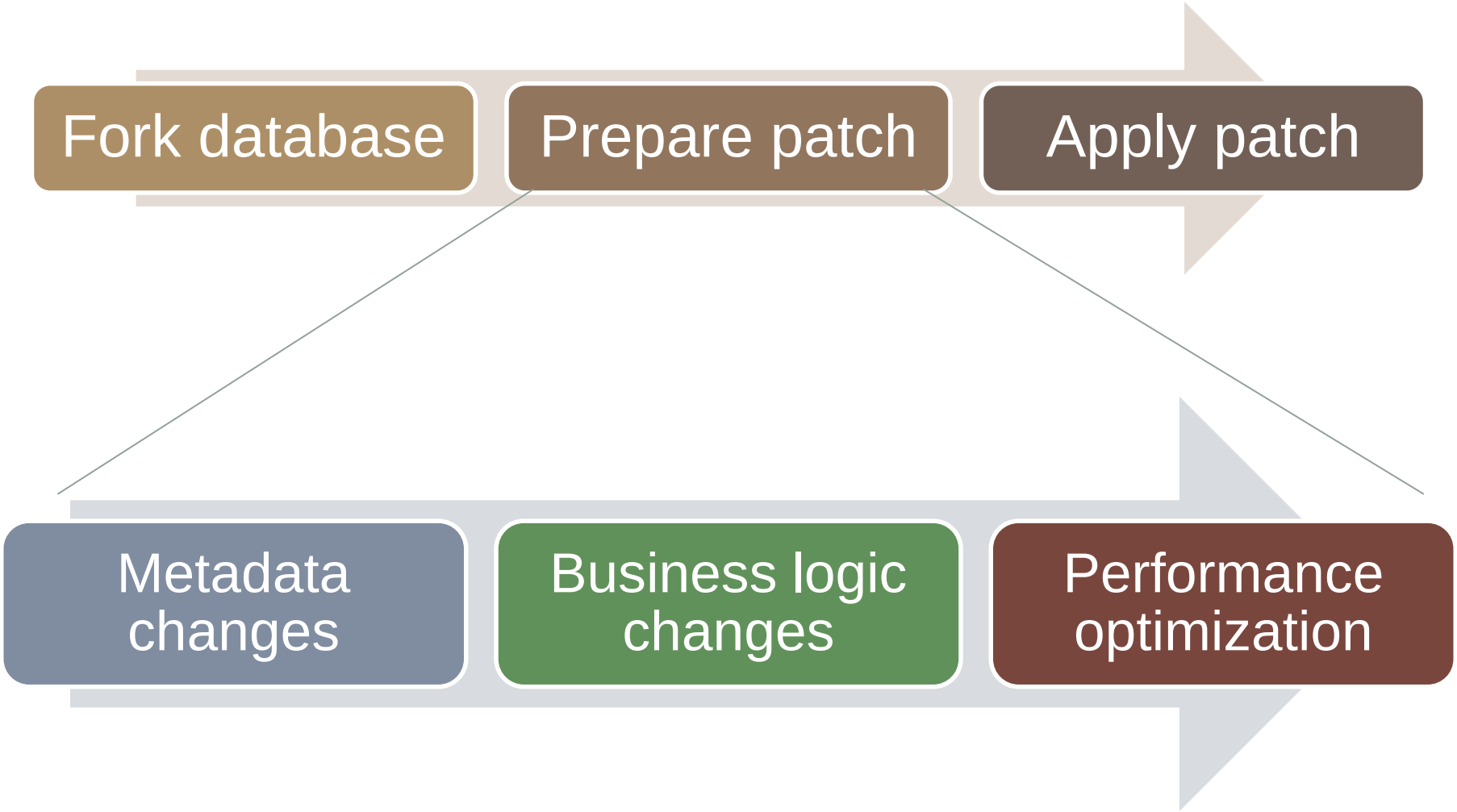
All together = SLOW.

Simple approach is impossible



The only available timeframe for switch to the new version – **Christmas & New Year** Eve

Special approach needed



Infrastructure for migration

- 2 servers (similar)
  - Production is 64Gb RAM, Dell Storage
  - Test is 32Gb RAM, RAID1
- Tools to verify the migration
  - SQL queries compatibility
  - SQL queries plans - performance

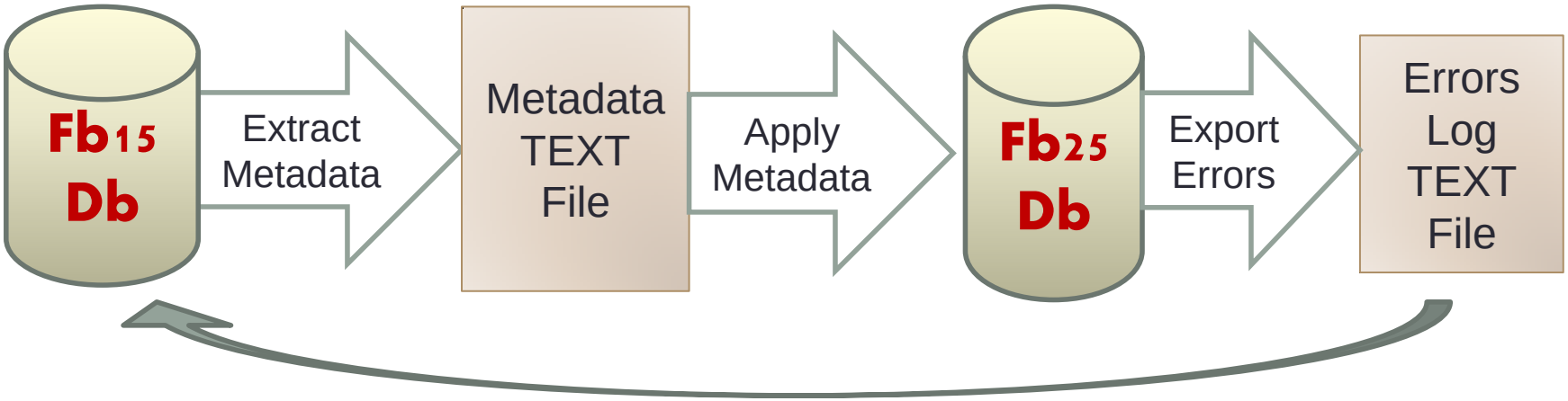
Steps

1. Prepare metadata
2. Test convert data to 2.5
3. Application migration
    1. Check SQL queries in applications for compatibility,
    2. Change SQLs if necessary
    3. Check execution plans of SQL queries
    4. Change plans if necessary
4. Test run
5. Final run

Step 1: Prepare metadata

1. Extract metadata from Firebird 1.5 to script

   Isql –x (or using GUI tools)
2. Run metadata script in Firebird 2.5
3. Get output as “errors.txt” and analyze it
4. Patch 1.5 database (should be compatible with 1.5 and 2.5), and external script



Errors in pure metadata – part 1

1. **Ambiguous field name** between table X and table Y (need to use aliases!) ~40%
2. Data type unknown. **Blob sub_types bigger than 1** (text) are for internal use only. (wrong sub_type in BLOB definitions) ~10%
3. Attempt to update read-only column (changes in AFTER UPDATE/AFTER DELETE triggers) ~7%

Errors in pure metadata – part 2

4. In 1.5 – function TRIM was in UDF, now it's embedded function with different parameters ~5%
5. Expression evaluation not supported. **Strings cannot be multiplied** in dialect 3 (explicit casting needed). ~1%
6. New Keywords (GLOBAL)
7. Other errors ~1%

Summary of Step 1

- 800+ errors
- 2 weeks to fix
- Scripts:
  - Fixes in existing Firebird 1.5 database
  - **Patch25.sql** - External script to patch 2.5 database

Next: we are ready to convert data to Firebird 2.5

Step 2: Test convert data to 2.5

1. Backup patched 1.5 database
2. Restore with **–fix_fss_metadata** and –fix_fss_data options
3. Apply Patch25.sql script to restored database
4. Backup/restore under 2.5

Backup – 2 hours, restore - 6 hours.

Summary of step 2

- 2 days
- We have database in 100% Firebird 2.5 compatible format

- Next: need to check and change SQL queries in all our applications

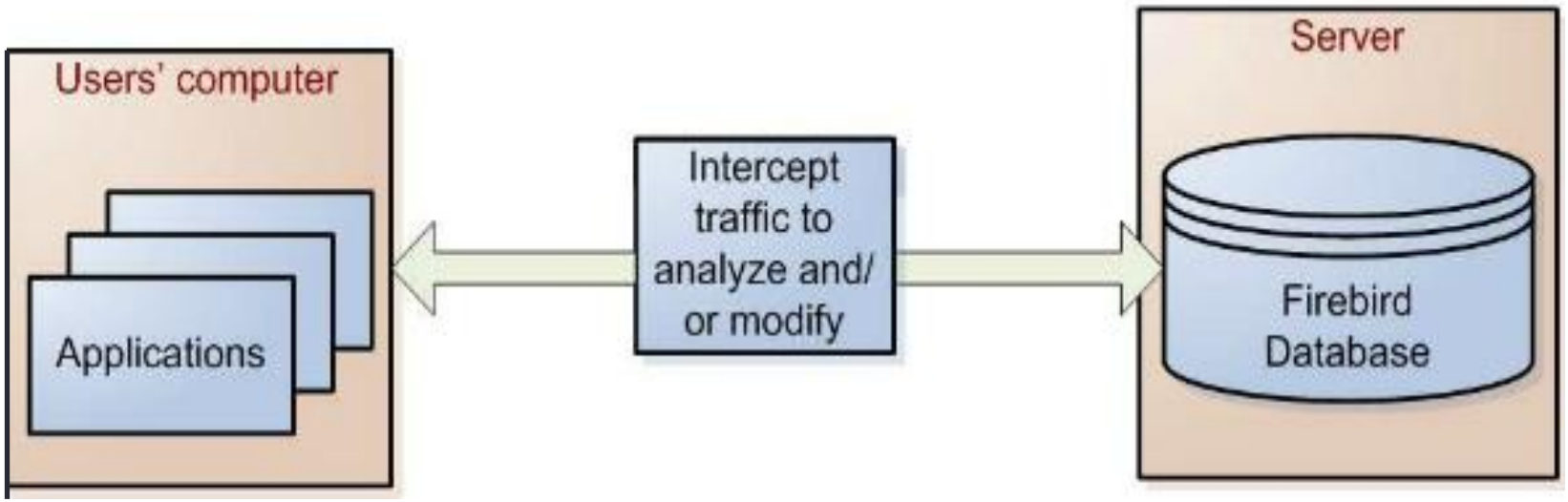
Step 3: Application migration

1. Check SQL queries in applications for compatibility,
2. Change SQLs if necessary
3. Check execution plans of SQL queries
4. Change plans if necessary

This is the most complex step!

How to check SQL queries for compatibility

- We needed to **log** all SQL queries from **all applications**
  - Own-written applications
  - Closed-source application

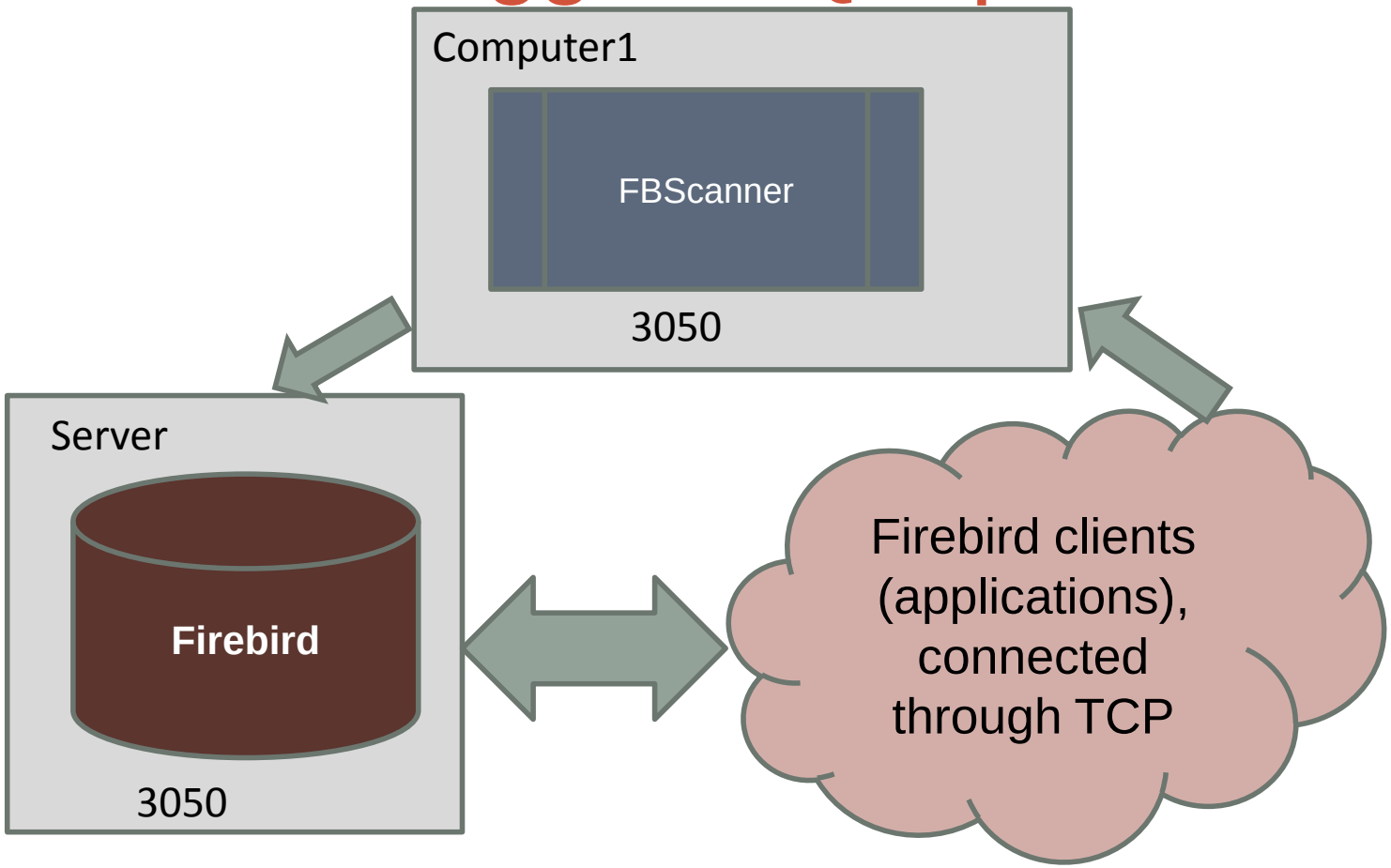


Tip: logging is complicated

- **MON$** tables does not help – they are snapshots (and in 2.1+ only) and make the heavy load
- **FBTrace API** is 2.5+ only

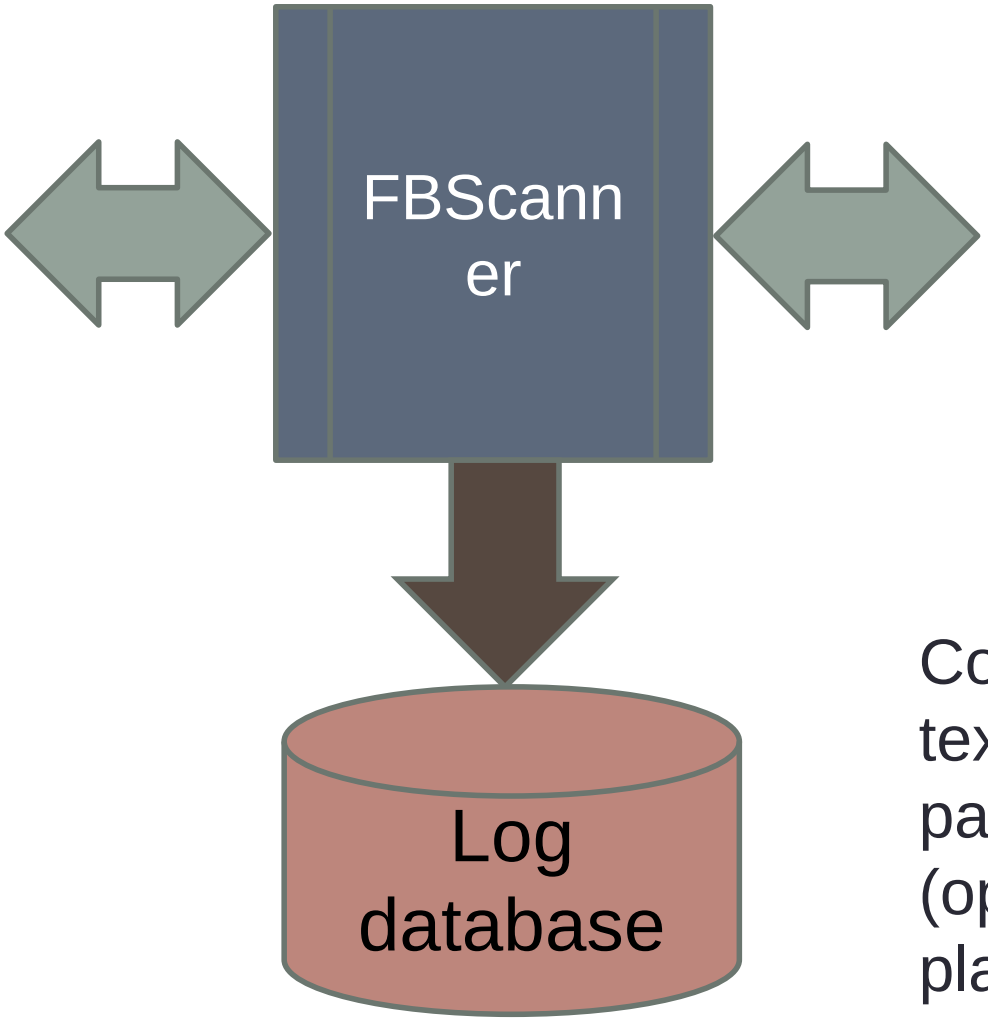
- **FBScanner** is IBSurgeon's commercial tool which analyze the network traffic and stores the full log, it's the only true logging solution

FBScanner logged SQL queries

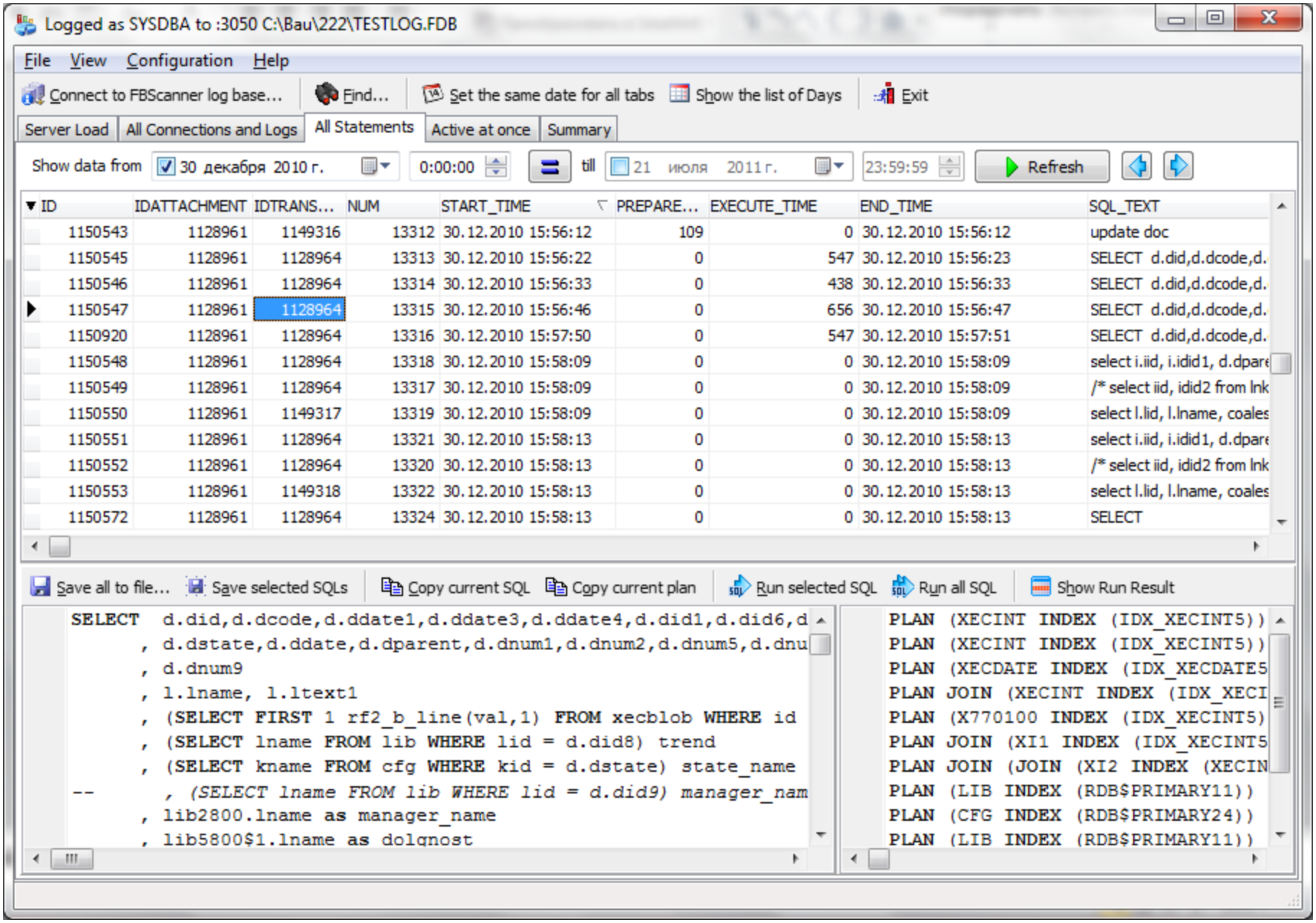


Several workstations were run through FBScanner one by one to reduce performance impact and make log more “linear”.

All SQL queries were stored to the log



Complete SQL texts with parameters and (optionally) plans!


Logged as SYSDBA to :3050 C:\Bau\222\TESTLOG.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
Show data from
30 декабря 2010 г.
0:00:00
till
21 июля 2011 г.
23:59:59
Refresh
ID IDATTACHMENT IDTRANS... NUM START_TIME PREPARE... EXECUTE_TIME END_TIME SQL_TEXT
1150543 1128961 1149316 13312 30.12.2010 15:56:12 109 0 30.12.2010 15:56:12 update doc
1150545 1128961 1128964 13313 30.12.2010 15:56:22 0 547 30.12.2010 15:56:23 SELECT d.did,d.dcode,d.
1150546 1128961 1128964 13314 30.12.2010 15:56:33 0 438 30.12.2010 15:56:33 SELECT d.did,d.dcode,d.
1150547 1128961 1128964 13315 30.12.2010 15:56:46 0 656 30.12.2010 15:56:47 SELECT d.did,d.dcode,d.
1150920 1128961 1128964 13316 30.12.2010 15:57:50 0 547 30.12.2010 15:57:51 SELECT d.did,d.dcode,d.
1150548 1128961 1128964 13318 30.12.2010 15:58:09 0 0 30.12.2010 15:58:09 select i.iid, i.idid1, d.dpare
1150549 1128961 1128964 13317 30.12.2010 15:58:09 0 0 30.12.2010 15:58:09 /* select iid, idid2 from lnk
1150550 1128961 1149317 13319 30.12.2010 15:58:09 0 0 30.12.2010 15:58:09 select l.lid, l.lname, coales
1150551 1128961 1128964 13321 30.12.2010 15:58:13 0 0 30.12.2010 15:58:13 select i.iid, i.idid1, d.dpare
1150552 1128961 1128964 13320 30.12.2010 15:58:13 0 0 30.12.2010 15:58:13 /* select iid, idid2 from lnk
1150553 1128961 1149318 13322 30.12.2010 15:58:13 0 0 30.12.2010 15:58:13 select l.lid, l.lname, coales
1150572 1128961 1128964 13324 30.12.2010 15:58:13 0 0 30.12.2010 15:58:13 SELECT
Save all to file...
Save selected SQLs
Copy current SQL
Copy current plan
Run selected SQL
Run all SQL
Show Run Result
SELECT d.did,d.dcode,d.ddate1,d.ddate3,d.ddate4,d.did1,d.did6,d
, d.dstate,d.ddate,d.dparent,d.dnum1,d.dnum2,d.dnum5,d.dnu
, d.dnum9
, l.lname, l.ltext1
, (SELECT FIRST 1 rf2_b_line(val,1) FROM xecblob WHERE id
, (SELECT lname FROM lib WHERE lid = d.did8) trend
, (SELECT kname FROM cfg WHERE kid = d.dstate) state_name
-- , (SELECT lname FROM lib WHERE lid = d.did9) manager_nam
, lib2800.lname as manager_name
, lib5800$1.lname as dolgnost
PLAN (XECINT INDEX (IDX_XECINT5))
PLAN (XECINT INDEX (IDX_XECINT5))
PLAN (XECDATE INDEX (IDX_XECDATE5
PLAN JOIN (XECINT INDEX (IDX_XECI
PLAN (X770100 INDEX (IDX_XECINT5)
PLAN JOIN (XI1 INDEX (IDX_XECINT5
PLAN JOIN (JOIN (XI2 INDEX (XECIN
PLAN (LIB INDEX (RDB$PRIMARY11))
PLAN (CFG INDEX (RDB$PRIMARY24))
PLAN (LIB INDEX (RDB$PRIMARY11))

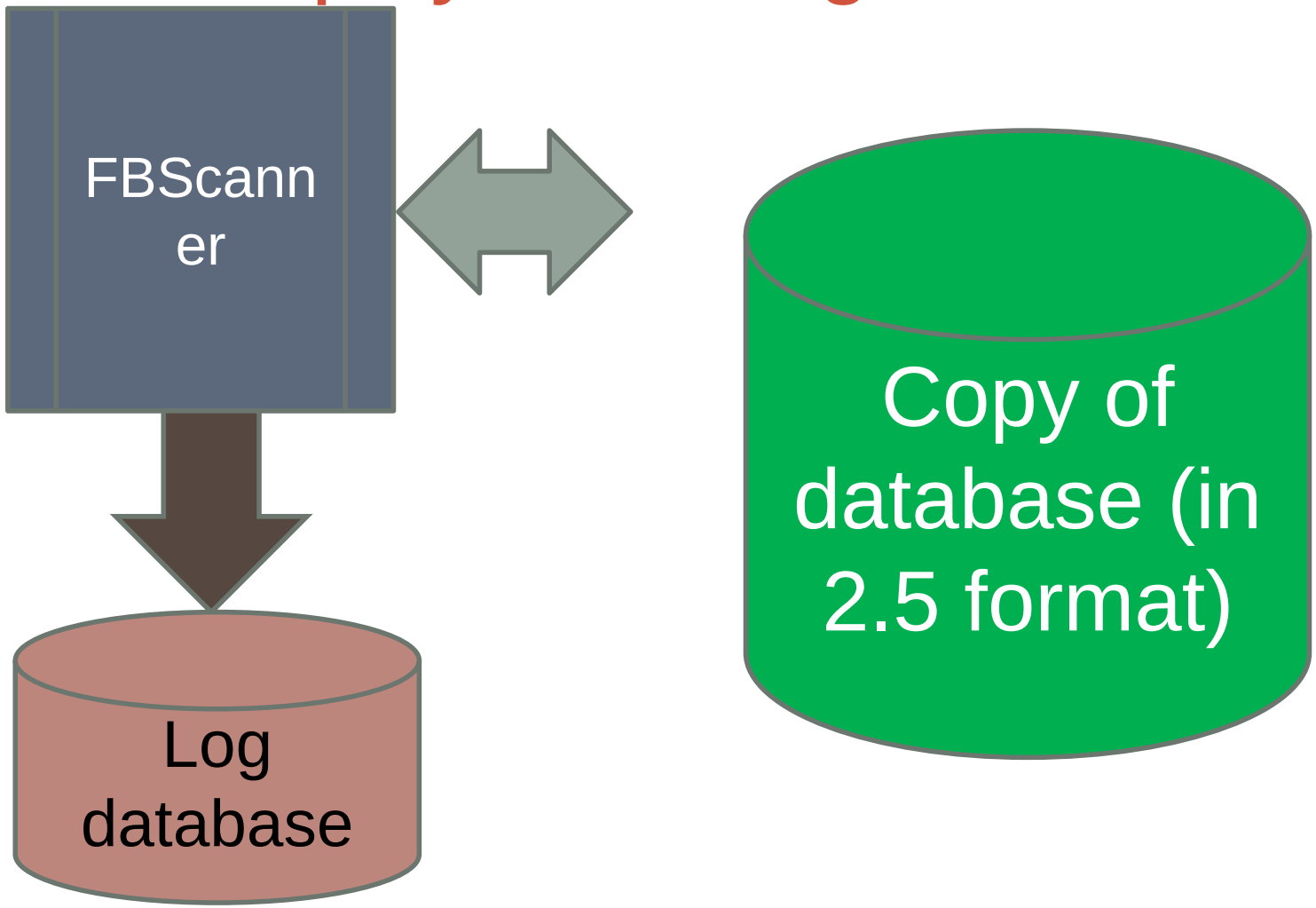
To find incompatible SQLs

We need to **“play” log** to the copy of 2.5 database

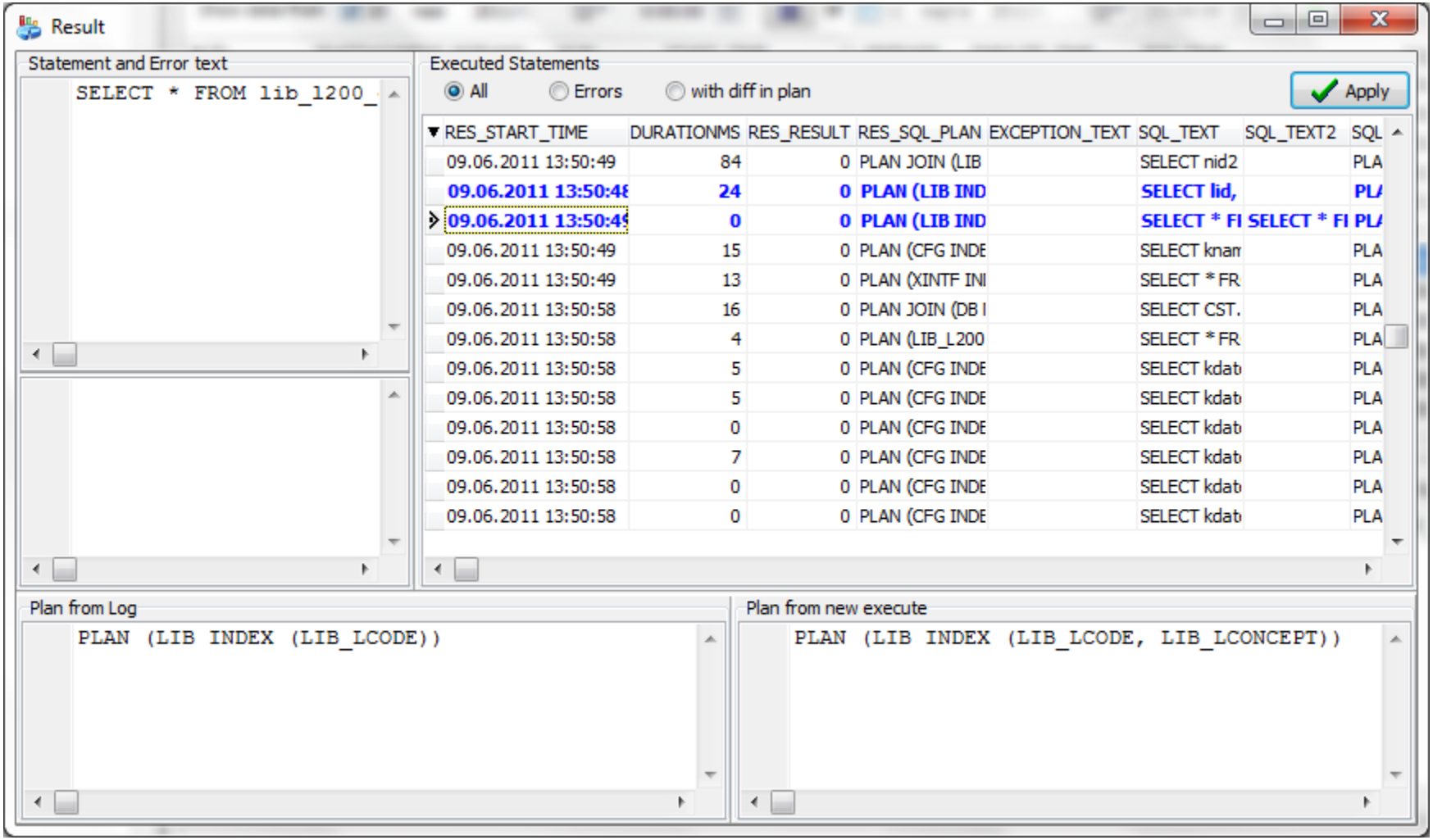
- Make prepare
- Try to execute
- Catch exceptions/errors if any
- Log SQL execution plans
- Log SQL execution times

Find the differences in plans and delays

FBScanner played the log



Differences in plans and execution times between 1.5 and 2.5 are highlighted

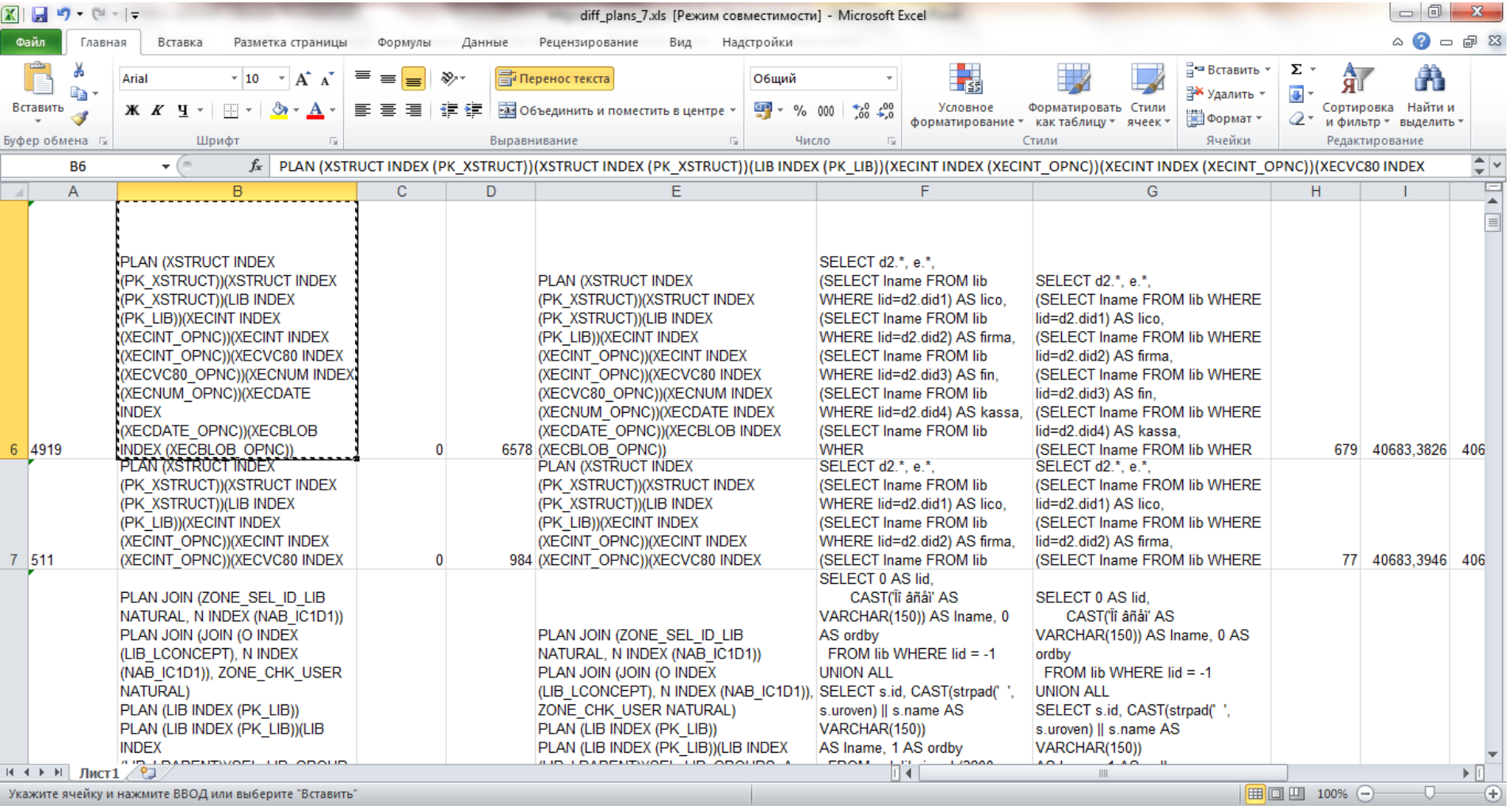


Play log results

- We had ~55000 queries to analyze
- Only 280 has different plans
- ~400 has slower Execution time at 2.5 than 1.5
- ~50 queries raised exceptions

Only ~750 queries from 55000 required investigation.

Sorted log was exported to Excel



	A	B	C	D	E	F	G	H	I	
6	4919	PLAN (XSTRUCT INDEX (PK_XSTRUCT))(XSTRUCT INDEX (PK_XSTRUCT))(LIB INDEX (PK_LIB))(XECINT INDEX (XECINT_OPNC))(XECINT INDEX (XECINT_OPNC))(XECVC80 INDEX (XECVC80_OPNC))(XECNUM INDEX (XECNUM_OPNC))(XECDATE INDEX (XECDATE_OPNC))(XECBLOB INDEX (XECBLOB_OPNC))	0	6578	PLAN (XSTRUCT INDEX (PK_XSTRUCT))(XSTRUCT INDEX (PK_XSTRUCT))(LIB INDEX (PK_LIB))(XECINT INDEX (XECINT_OPNC))(XECINT INDEX (XECINT_OPNC))(XECVC80 INDEX (XECVC80_OPNC))(XECNUM INDEX (XECNUM_OPNC))(XECDATE INDEX (XECDATE_OPNC))(XECBLOB INDEX (XECBLOB_OPNC))	SELECT d2.*, e.*, (SELECT lname FROM lib WHERE lid=d2.did1) AS lico, (SELECT lname FROM lib WHERE lid=d2.did2) AS firma, (SELECT lname FROM lib WHERE lid=d2.did3) AS fin, (SELECT lname FROM lib WHERE lid=d2.did4) AS kassa, (SELECT lname FROM lib WHER	SELECT d2.*, e.*, (SELECT lname FROM lib WHERE lid=d2.did1) AS lico, (SELECT lname FROM lib WHERE lid=d2.did2) AS firma, (SELECT lname FROM lib WHERE lid=d2.did3) AS fin, (SELECT lname FROM lib WHERE lid=d2.did4) AS kassa, (SELECT lname FROM lib WHER	679	40683,3826	406
7	511	PLAN (XSTRUCT INDEX (PK_XSTRUCT))(XSTRUCT INDEX (PK_XSTRUCT))(LIB INDEX (PK_LIB))(XECINT INDEX (XECINT_OPNC))(XECINT INDEX (XECINT_OPNC))(XECVC80 INDEX	0	984	PLAN (XSTRUCT INDEX (PK_XSTRUCT))(XSTRUCT INDEX (PK_XSTRUCT))(LIB INDEX (PK_LIB))(XECINT INDEX (XECINT_OPNC))(XECINT INDEX (XECINT_OPNC))(XECVC80 INDEX	SELECT d2.*, e.*, (SELECT lname FROM lib WHERE lid=d2.did1) AS lico, (SELECT lname FROM lib WHERE lid=d2.did2) AS firma, (SELECT lname FROM lib	SELECT d2.*, e.*, (SELECT lname FROM lib WHERE lid=d2.did1) AS lico, (SELECT lname FROM lib WHERE lid=d2.did2) AS firma, (SELECT lname FROM lib WHERE	77	40683,3946	406
		PLAN JOIN (ZONE_SEL_ID_LIB NATURAL, N INDEX (NAB_IC1D1)) PLAN JOIN (JOIN (O INDEX (LIB_LCONCEPT), N INDEX (NAB_IC1D1)), ZONE_CHK_USER NATURAL) PLAN (LIB INDEX (PK_LIB)) PLAN (LIB INDEX (PK_LIB))(LIB INDEX			PLAN JOIN (ZONE_SEL_ID_LIB NATURAL, N INDEX (NAB_IC1D1)) PLAN JOIN (JOIN (O INDEX (LIB_LCONCEPT), N INDEX (NAB_IC1D1)), ZONE_CHK_USER NATURAL) PLAN (LIB INDEX (PK_LIB)) PLAN (LIB INDEX (PK_LIB))(LIB INDEX	SELECT 0 AS lid, CAST('Îï âñåì' AS VARCHAR(150)) AS lname, 0 AS ordby FROM lib WHERE lid = -1 UNION ALL SELECT s.id, CAST(strpad(' ', s.uroven) || s.name AS VARCHAR(150)) AS lname, 1 AS ordby	SELECT 0 AS lid, CAST('Îï âñåì' AS VARCHAR(150)) AS lname, 0 AS ordby FROM lib WHERE lid = -1 UNION ALL SELECT s.id, CAST(strpad(' ', s.uroven) || s.name AS VARCHAR(150))			

Plan problem example (simplified)

```
select agb.eid, agb.kollast, agb.eid5, agb.first_eid,
agb.did3, agb.did4, agb.marker, agb.place, agb.mnp
from p101_ant_goods_balance agb
where agb.ddate <= :i$ddate and agb.l800 = :i$l800 and
agb.did3 = rf2_abs(3409) and agb.marker = :i$marker and
agb.did <> :i$did
```

- 2.5 PLAN (AGB INDEX (IDX$AGB_L800_DDDATE_ECON_EMP))
- 1.5 PLAN (AGB INDEX (IDX$AGB_L800_DDDATE_ECON_EMP, **IX$AGB_DID3**))

Another problems

- New Firebird keywords in closed-sourced software
  - Was hacked ☺
- UDF rFunc for 64-bit Windows and Linux
  - AUDFL https://www.assembla.com/wiki/show/audfl_rfunc

Results of migration

- At Firebird 2.5
  - Better performance in queries
  - Fast garbage collection (20 minutes instead of 1.5 hours)
  - Backup time reduced (1.5 hour instead of 2 hours)
  - Better use of multi-CPU hardware
  - 64 bit version of Firebird available
  - EXECUTION STATEMENT and other SQL features

Optimization Pack “Rodizio” offer

- To optimize databases we offer 1 year subscription to install as many as you need FBScanner+IBAnalyst+IBTM instances
- This instances will run indefinitely (no restrictions)
- Should be activated
  - Silent bundle is also possible, requires signing VAR agreement

Today – EUR 1200 (reg EUR 1500)

Next presentation – tomorrow 14-00

- Nov 26, 14-00
- Supercharging Firebird production systems: transactions, garbage, maintenance

Thanks and Contacts

Blog

http://FirebirdSurgeon.blogspot.com

Twitter

http://Twitter.com/ibsurgeon

Web

http://www.ib-aid.com