HQBIRD 2020 — FASTER, EASIER AND POWERFUL FIREBIRD

Dmitry Kuzmenko

Alexey Kovyazin

www.ib-aid.com

Firebird Conference 2019

Berlin, 17-19 October

IBPhoenix

YOUR PREMIER SOURCE OF FIREBIRD SUPPORT

IBSurgeon

MOSCOW EXCHANGE

Fast Reports

Reporting must be fast!

IB Expert

REDSOFT

HQbird

HQBird
Advanced Firebird for Big Databases

- HQbird is the advanced distributive of Firebird (v2.5 and 3.0), which includes: native replication, improved performance, automation of backups, restores, sweeps, cloud backups, performance monitoring (SQL and transactions), tools to develop SQL and analyse database structure, and recovery tools.
- 3 editions:
  - Standard: monitoring, optimization, backups
  - Professional: Standard + Recovery+ SQL Studio
  - **Enterprise**: Professional + Replication + Encryption + Performance Enhancements

HQbird Enterprise 2020: main features

1) Multithread sweep, backup and restore — 2-6X faster!
2) Replication for Firebird 2.5 and 3.0: Asynchronous and synchronous
3) Engine performance improvements: REFETCH, TempCacheThreshold, Pool of prepared statements, Pool of ESOE
4) Security, Encryption and Authentication enhancements
    1) libRSA – UDR for signing and security/Firebird 3-4
    2) libCluster – authentication solution for ESOE (FB3)
    3) Encryption plugin
5) Performance reports
6) HQbird Control Center

HQBird to Firebird

- We contribute our features to Firebird
- Agreement with core Firebird Team
- Timeframe – our features goes to community Firebird in a year
- Example – Firebird 4.0 Replication improvements, gbak with encryption
- MT features - Firebird 4.*
- Other features – Firebird 5.*
- No backports for Community edition 2.5/3.0

HQbird: Admin Tools

- HQbird includes all necessary administrative tools:
  - HQbird FBDataGuard: automatic backups, restores, maintenance, sweep, cloud backup to FTP, etc, performance monitoring with reports, email alerts and recommendations
  - HQbird Database Analyst: tables and indices sizes and fragmentation, garbage records version analysis, indices depth and quality analysis, etc
  - HQbird Mon$Logger – analysis of MON$ tables, to see currently active queries, connections, transactions, and highlight the problematic items
  - FirstAID, IBBackupSurgeon – recovery tools
  - Firebird SQL Studio – development tool

1. MULTITHREAD SWEEP, BACKUP AND RESTORE

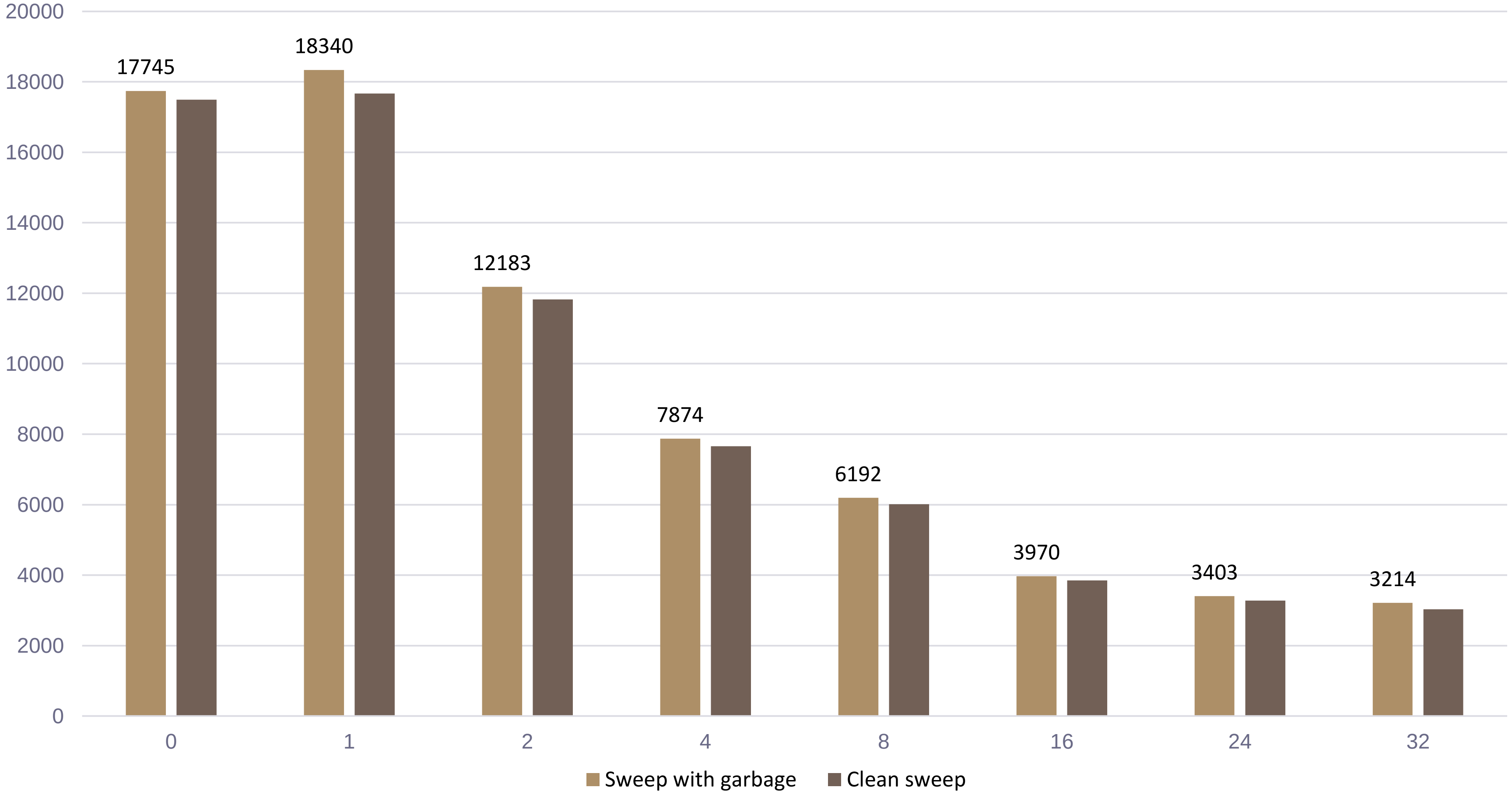
Multithread sweep in HQbird

- Command
  - gfix –sweep database –par NN
- Firebird.conf parameters
  - #MaxParallelWorkers = 1 (1-64)
  - #ParallelWorkers = 1
- Included in HQbird Enterprise 2020
  - Firebird 2.5 and 3.0

Graphs from test machine

- Xeon CPU with 16 cores/32 threads
- 64Gb
- RAID10 SAS15k
  - With SSD slightly faster than RAID10 SAS15k
  - With Enterprise HDD slightly slower than RAID10 SAS15k
  - The smallest advantage is on slow customer-grade HDDs

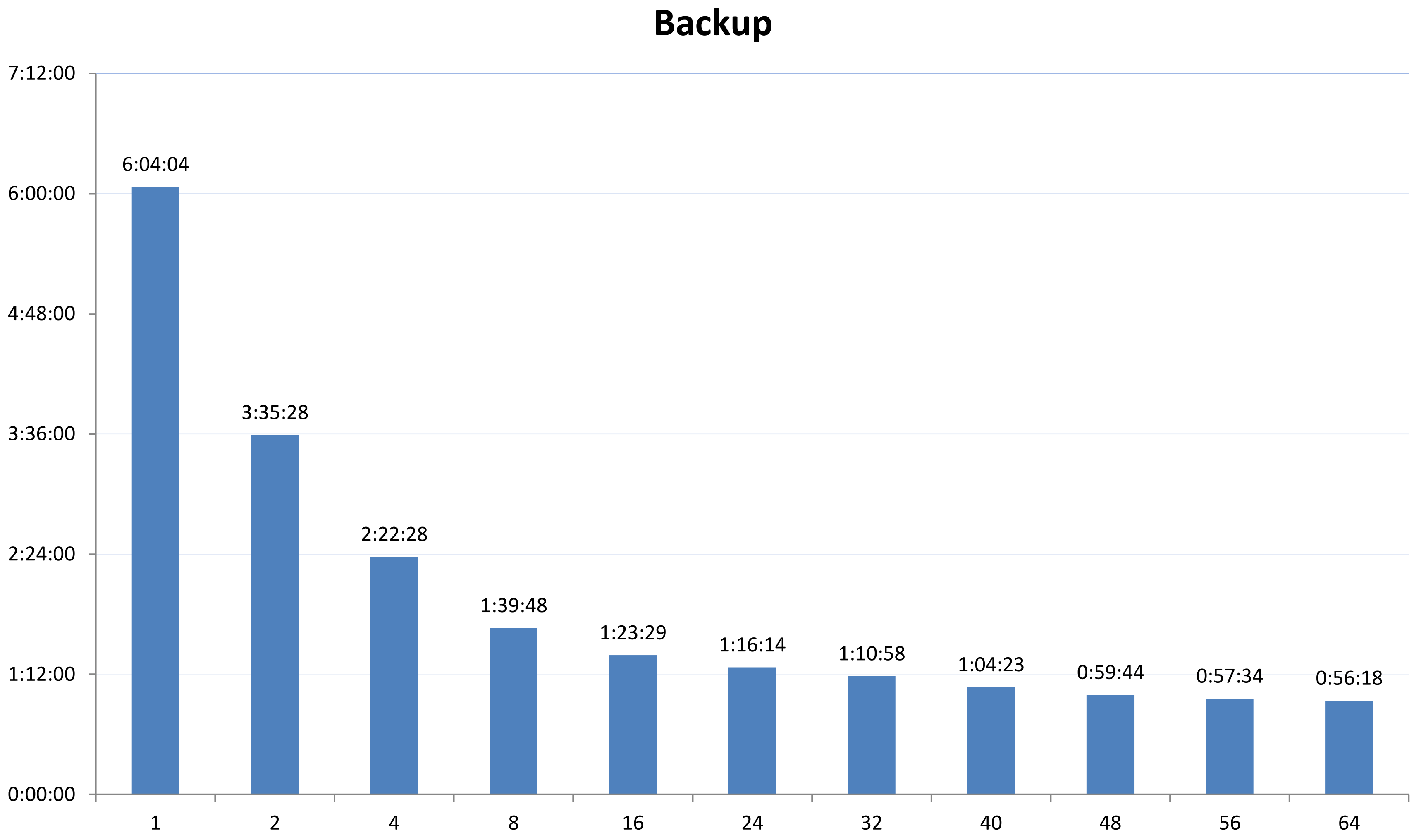
Multithread sweep: 4x faster



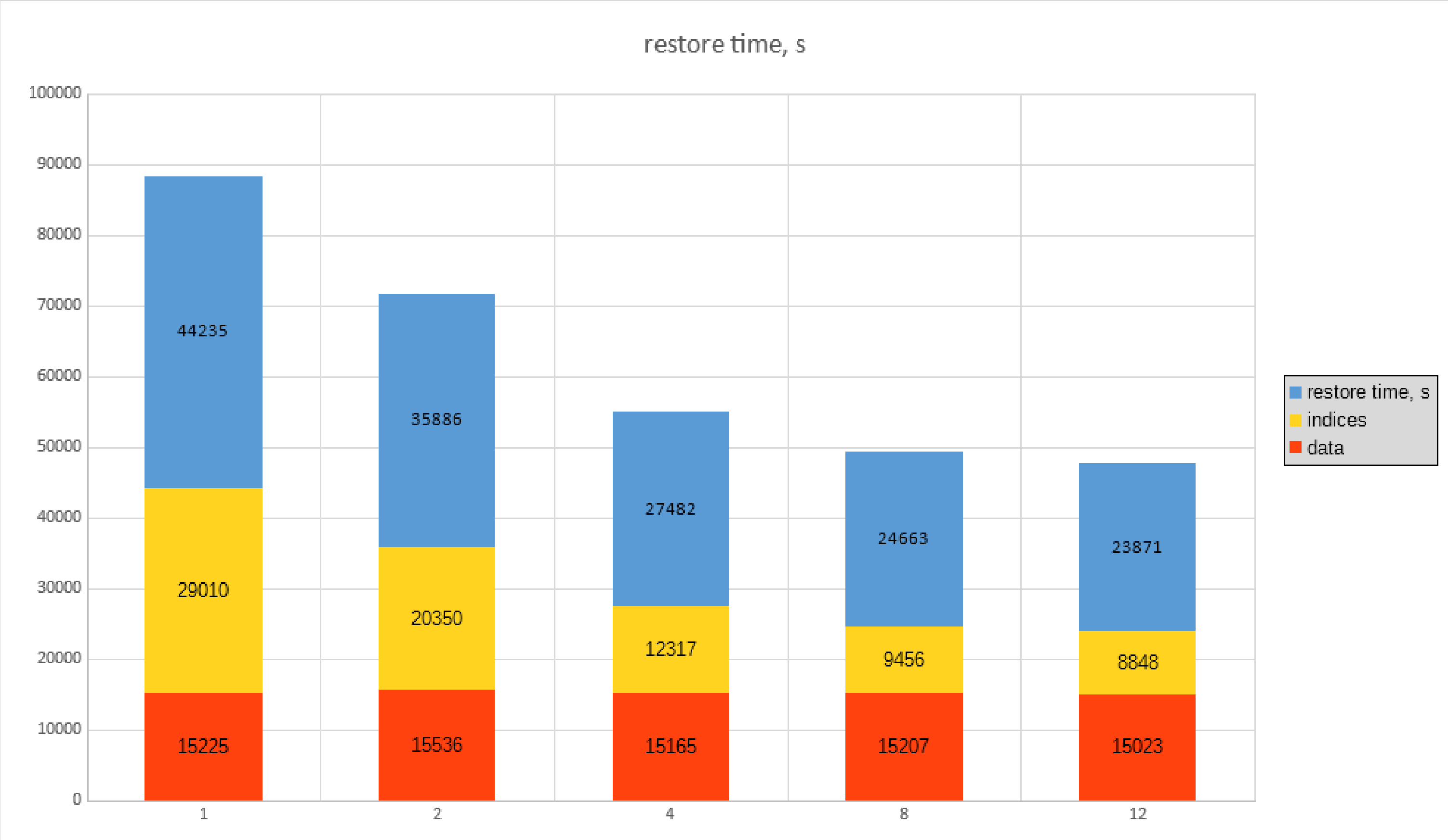
Multithread backup/restore

- Command
    - gbak –b –par NN database backup
    - gbak –c –par NN backup database
    - Services API also supported
- Backup format is 100% the same as usual

Multithread backup: 4-5x faster



Multithread restore: 2x faster



Only indices restore uses multiple cores, so overall improvement depends on the number/size of indices – the more you have the bigger will be improvement

Summary for multithread features

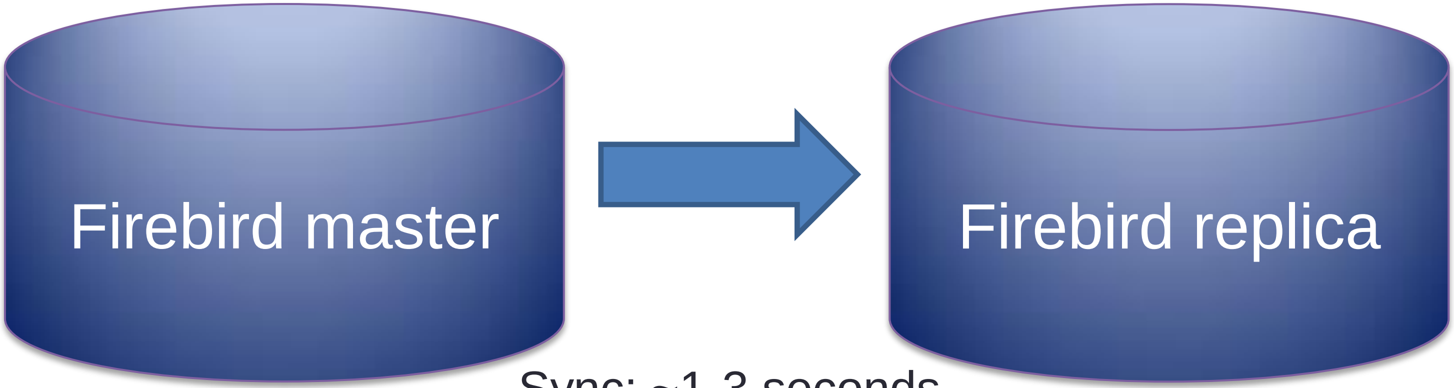
- Included in HQbird Enterprise 2020
- 100% compatible backup and restore with standard Firebird 2.5 and 3.0
- Will be available in standard Firebird 5+

2. REPLICATION FOR 2.5 AND 3.0

Why we need replication?

- If database size is big (20-100Gb)
    - Automatic recovery with FirstAID is very slow (downtime > 8 hours) or impossible (>100Gb)
    - Manual recovery is expensive (US$1000+) and slow (1 day+)
- Big database = Big company
    - Losses are not tolerated
    - Downtime is very expensive ($$$)
- Performance
    - Split read load between master and replica
    - Improve write speed

How replication works?



Sync: ~1-3 seconds
Async: from 10 sec

High Availability is based on native replication:

1. No Backup/restore needed
2. Easy setup (configuration only), no triggers
3. Logical replication, DML and DDL
4. Read-only replica for reports or analytics

High availability solutions in HQbird

1)Warm-standby

   1) Based on async replication, can be geographically split

   2) Requires 1 replica server, recommended for 99% solutions

2) Cloud backup

   1)Based on async replication

   2)Allows point in time recovery (partial)

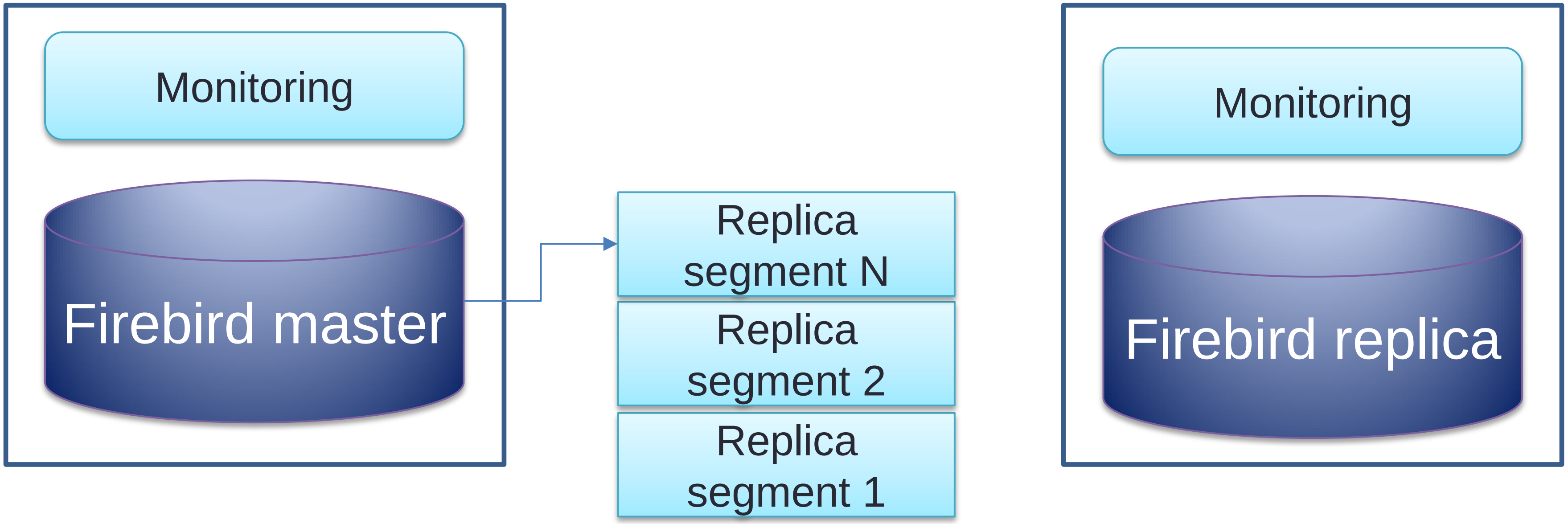
   3)Backup server can be anywhere in the cloud

3) Special deployment option (depends on the application)

   1) Based on sync replication, reduces downtime to several seconds

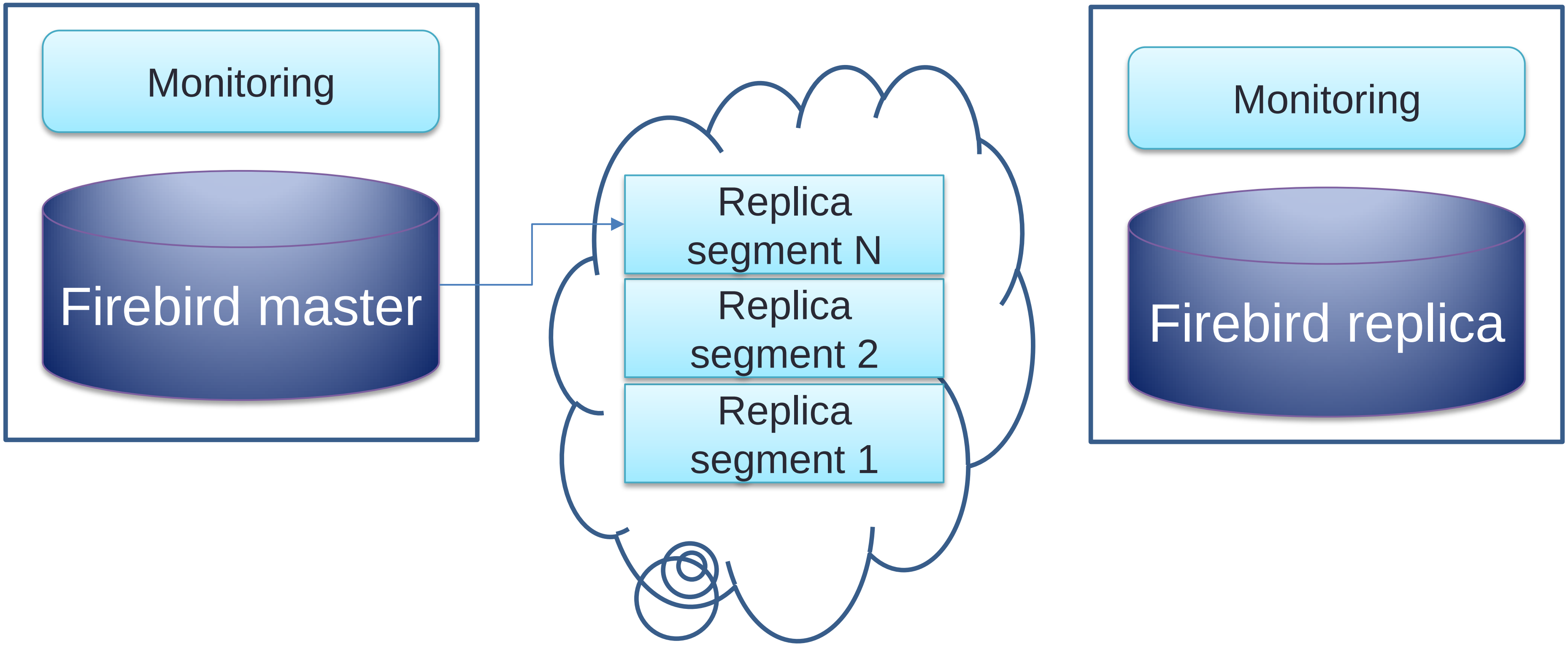
   2) Requires 2+ replica servers and high speed network connection (10Gb+ required)

Warm-standby

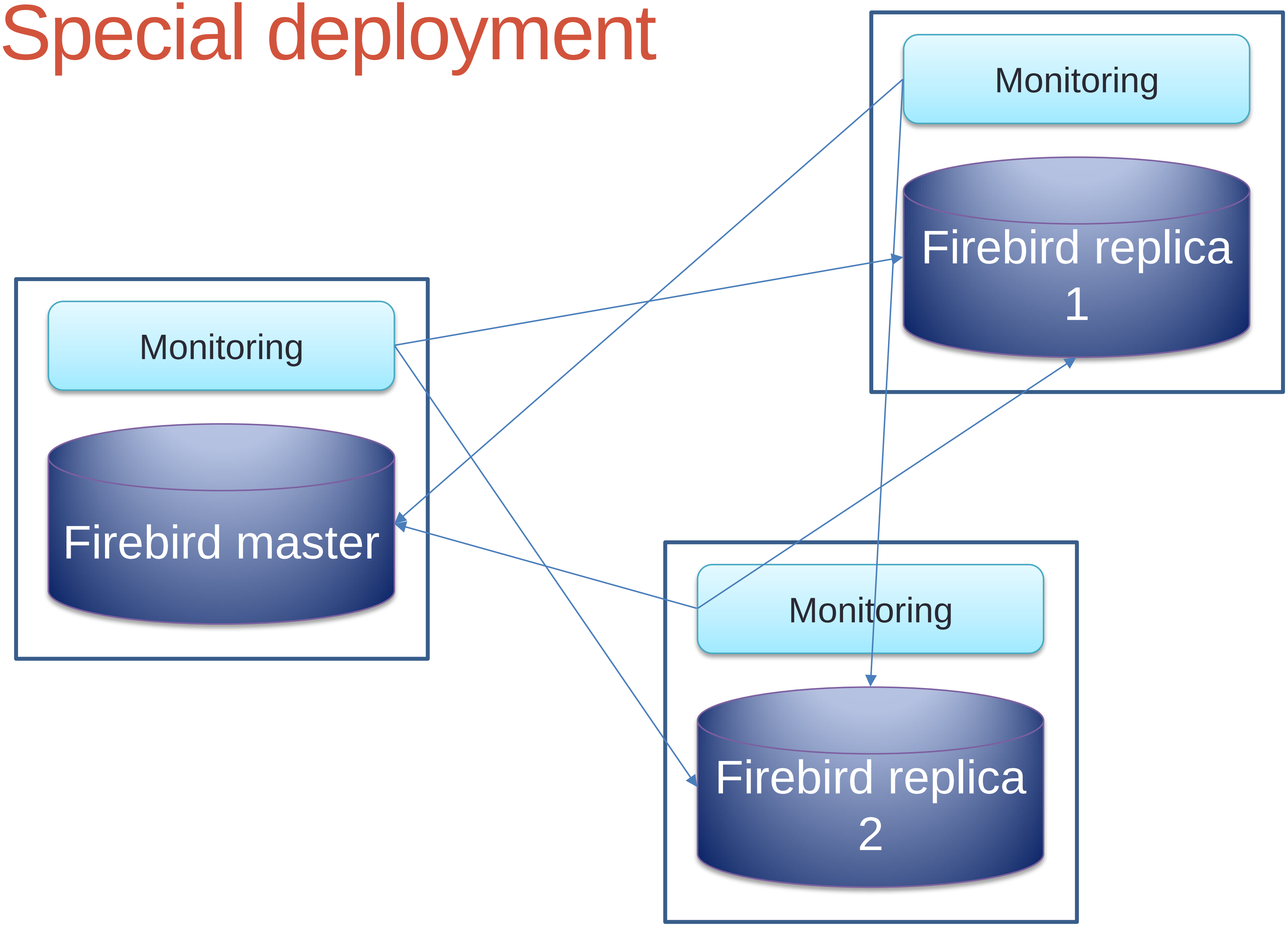


Interval can be set to 10-90 seconds

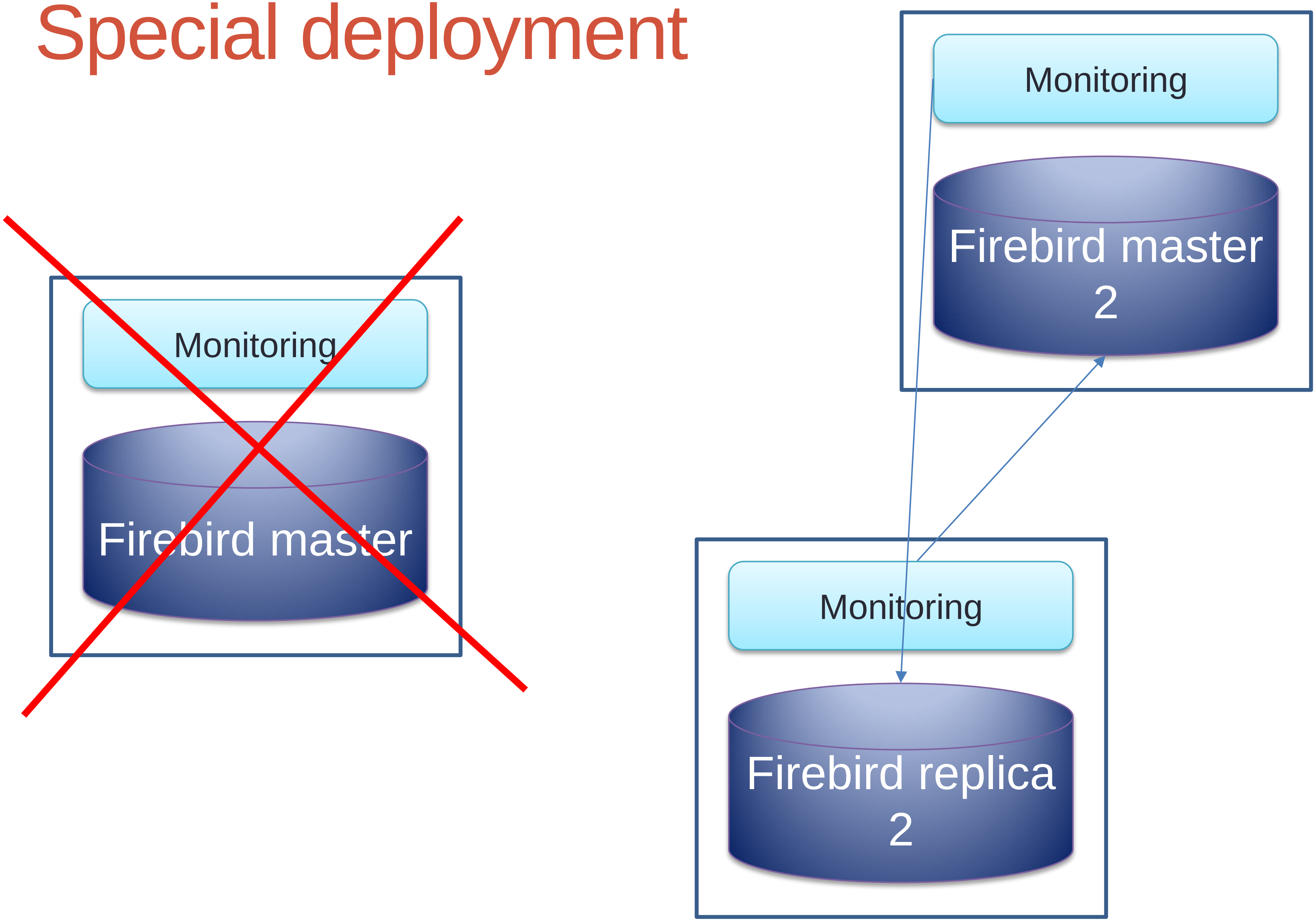
Cloud backup – FTP (Azure, Amazon, Google Cloud etc)



Special deployment



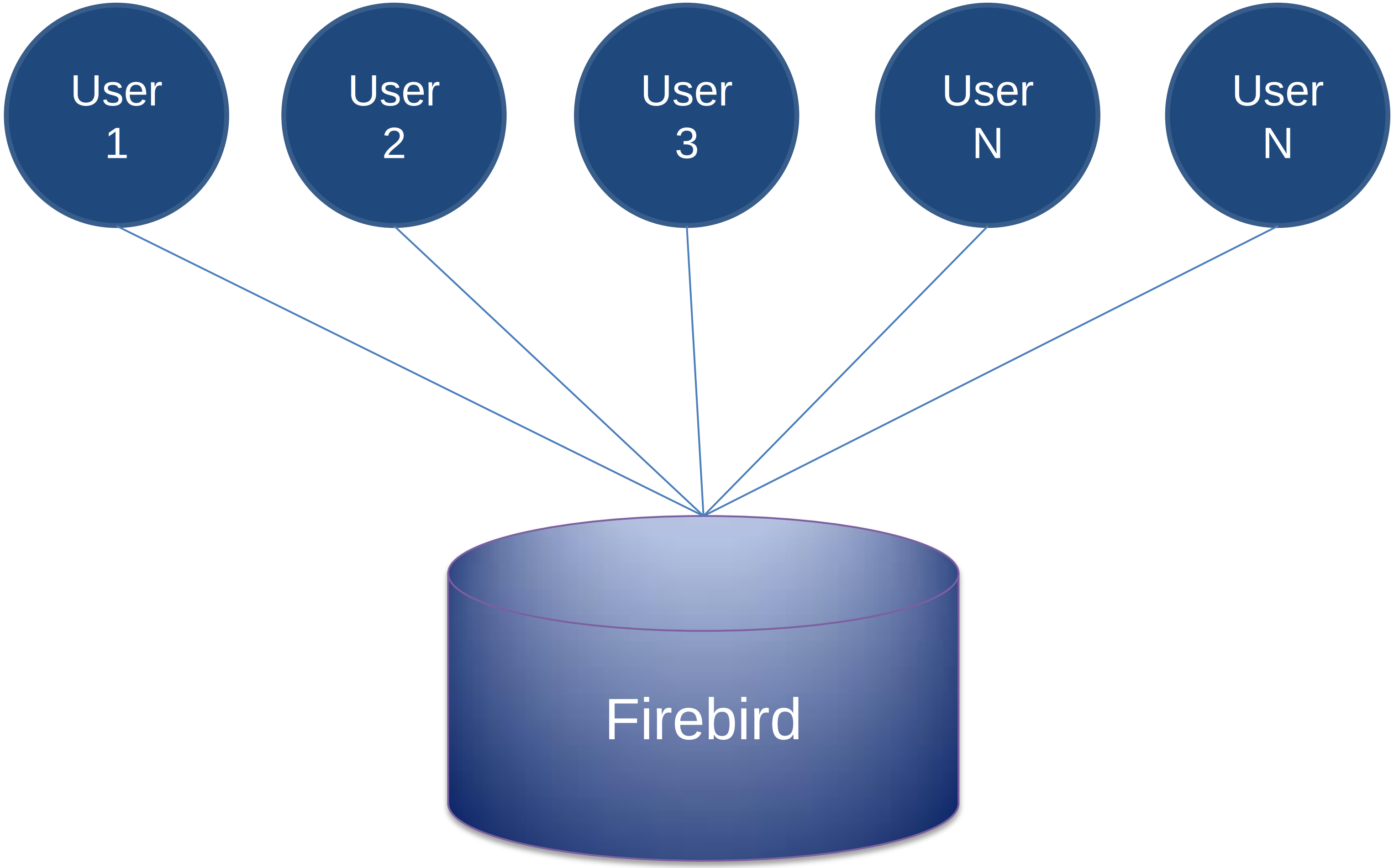
Special deployment



Replication in numbers

- What is the performance impact?
  - For async replication, around 1-2%
  - For sync replication, around 3-5%, depends on network speed
- How many users can be supported?
  - The largest client has 1200 daily users for 660Gb database
- What is the delay for replication
  - For async practical limit is 10 seconds
  - For sync — no delay, but decrease the performance of INSERT/UPDATE/DELETE

How to double Firebird performance with sync replication?

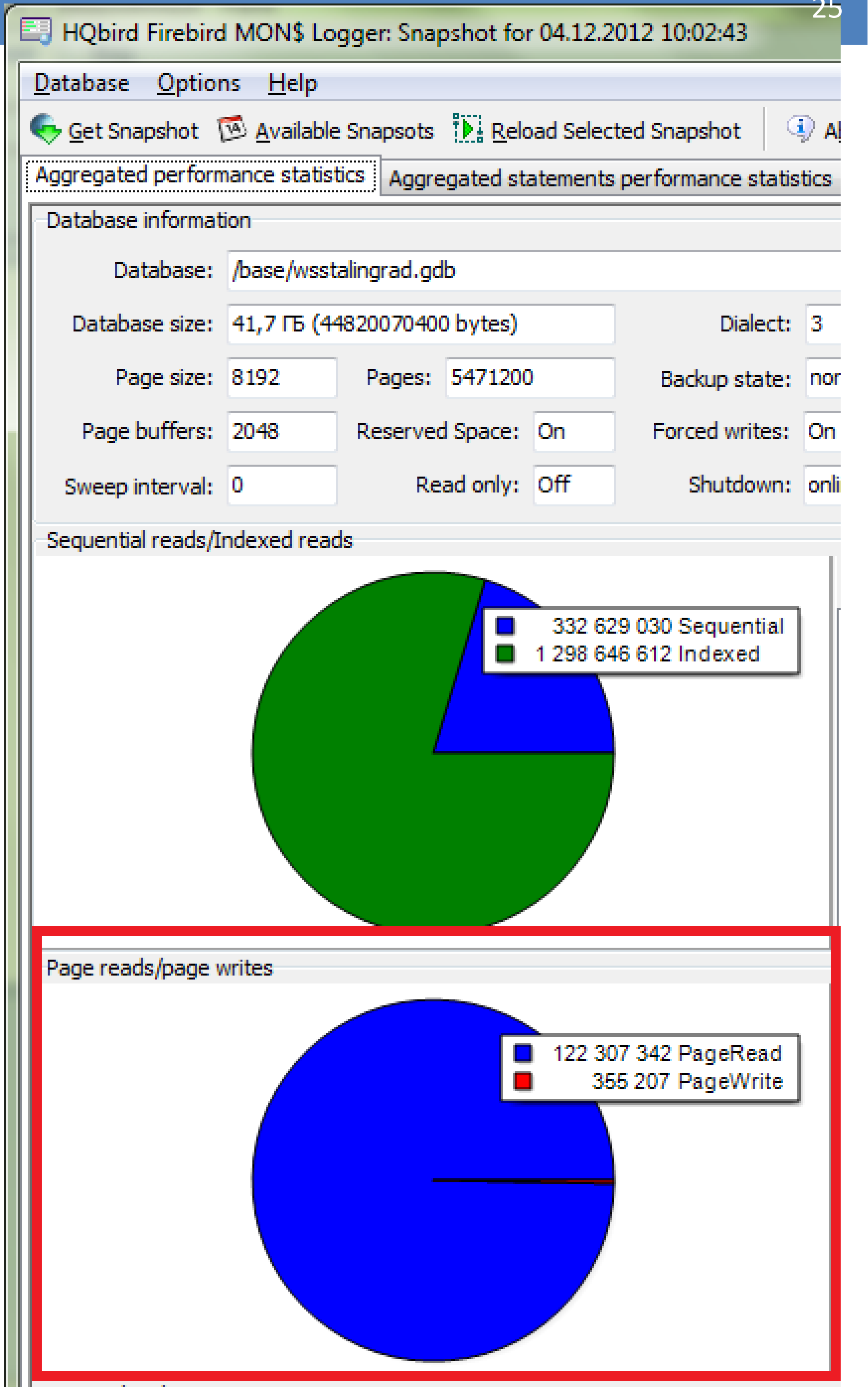


Part 1. Scale reads

Typical Reads/Writes ratio

=

99 to 1



Database: /base/wsstalingrad.gdb

Database size: 41,7 ГБ (44820070400 bytes)

Dialect: 3

Page size: 8192

Pages: 5471200

Backup state: nor

Page buffers: 2048

Reserved Space: On

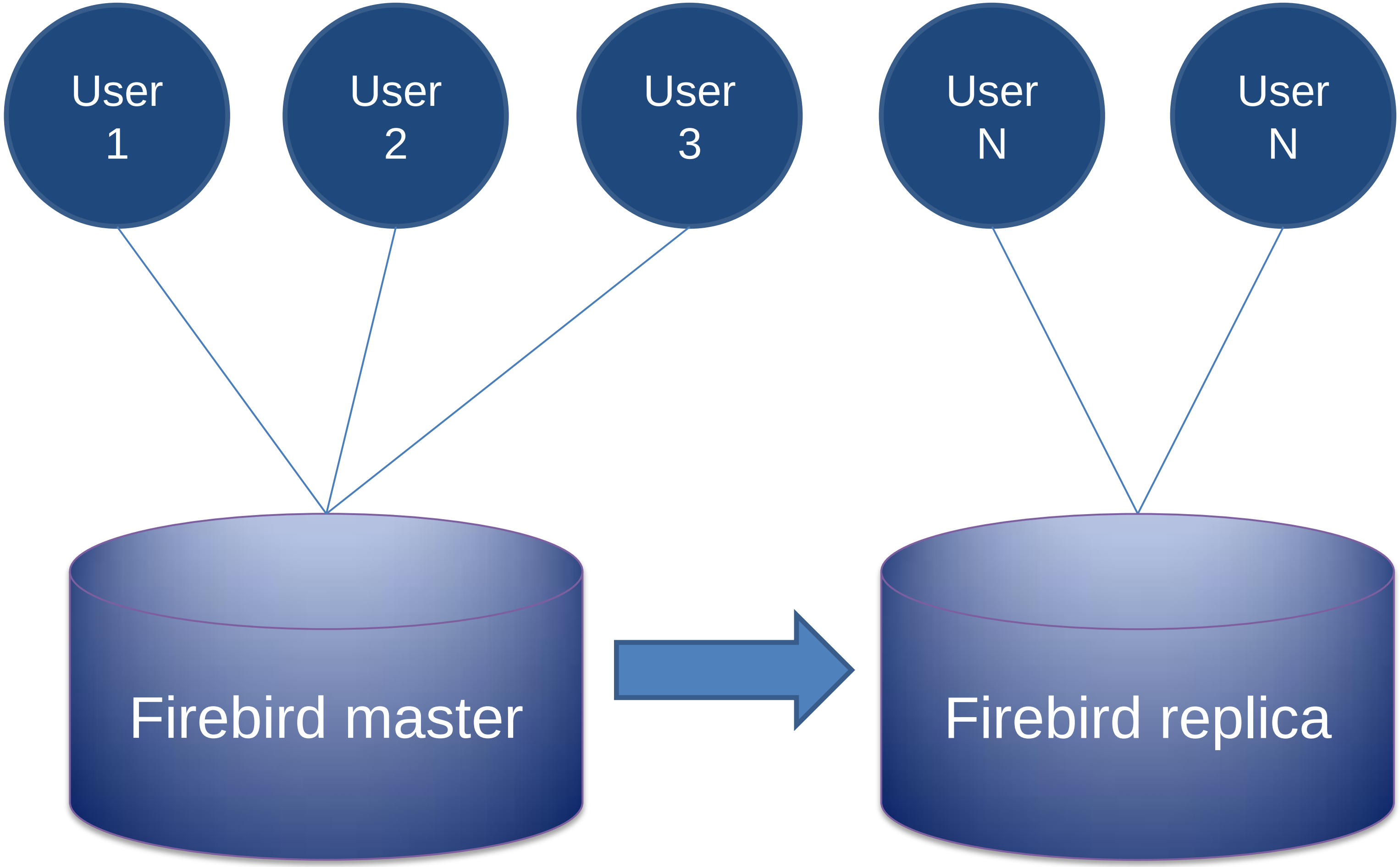
Forced writes: On

Sweep interval: 0

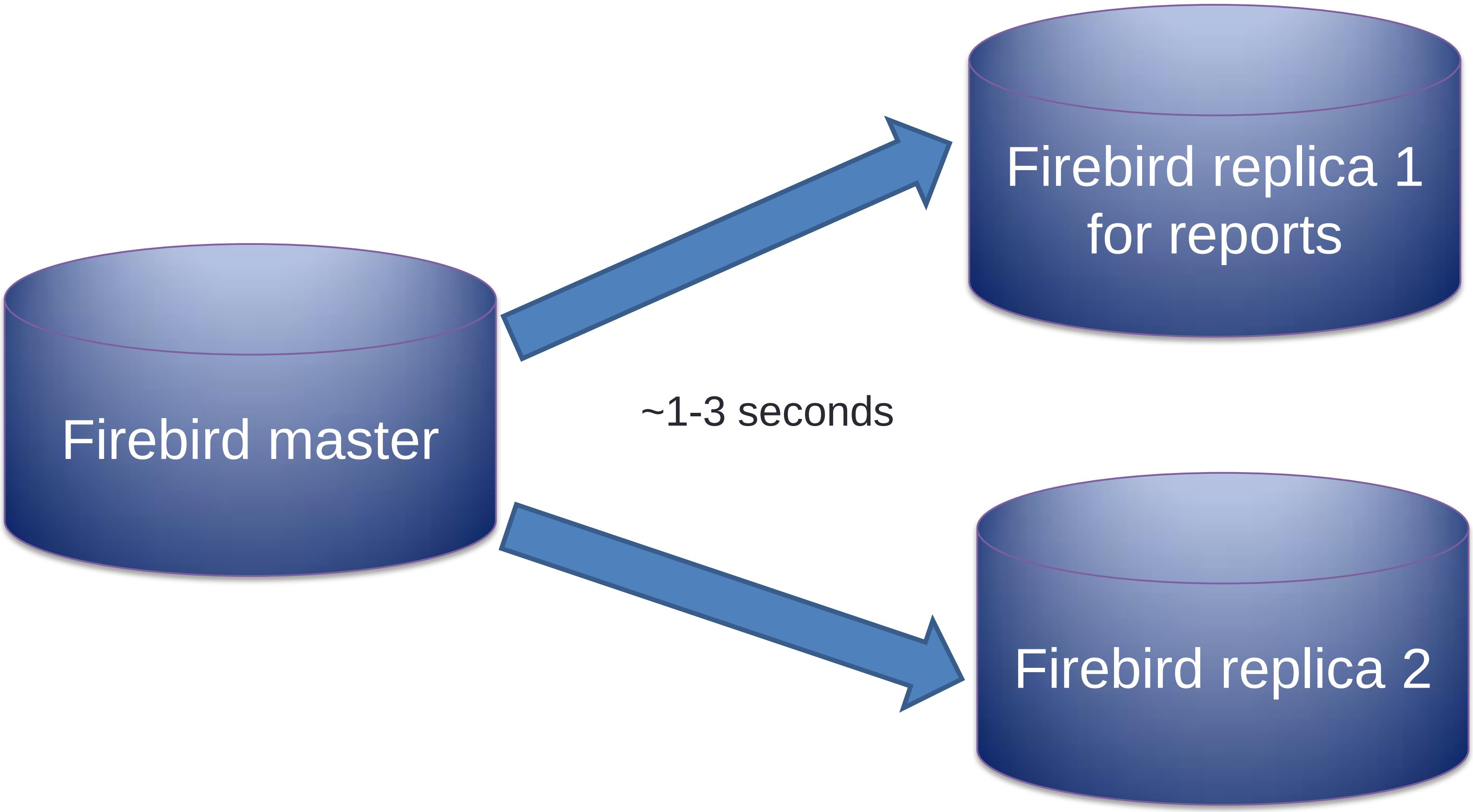
Read only: Off

Shutdown: onli

Split read load across master and replica



HQbird Enterprise allows several synchronized replicas



Key features of sync replication

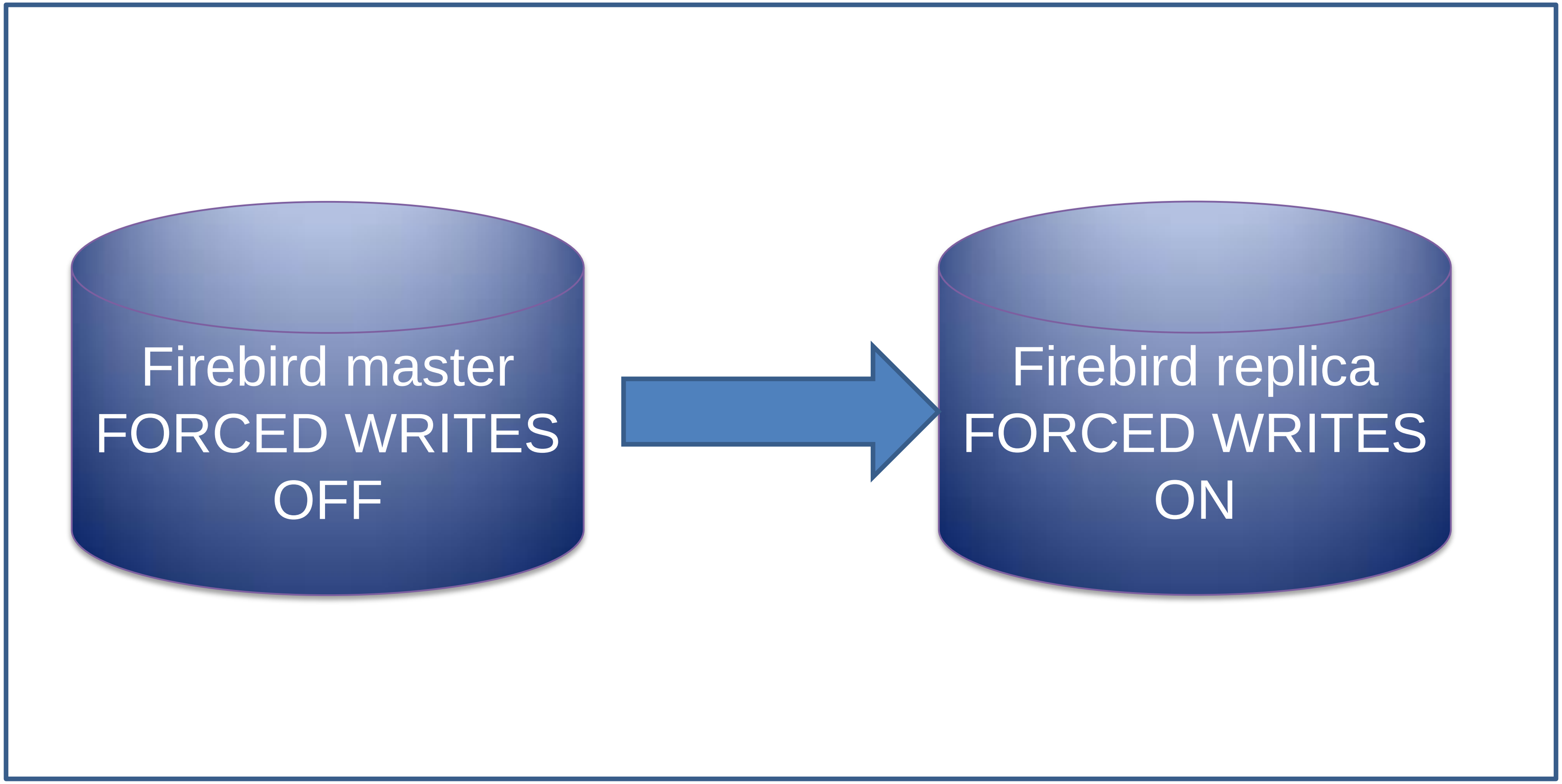
- Fast real-time replication of records immediately on commit
- Specific tables can be excluded from replica
- Easy setup
    - no triggers needed, no schema changes (only PK/UK for each table)
    - No backup/restore needed
    - Supports metadata changes (DDL)
- Support of Firebird 2.5 and 3.0

Part 2. Scale writes

- Forced Writes = OFF ?

Pros – Increases writes speed up to 3 times
Contra - Big chance of a corruption

Scale writes configuration



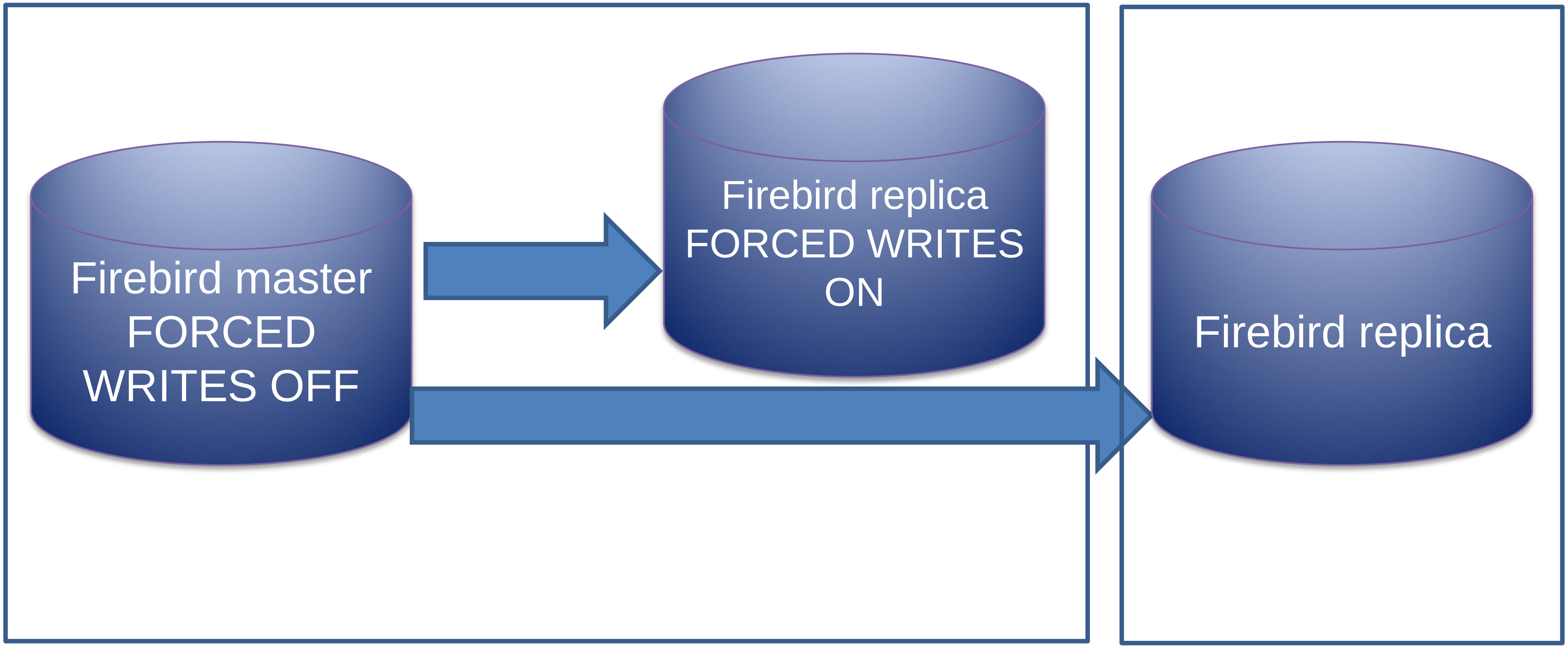
Firebird master
FORCED WRITES
OFF

Firebird replica
FORCED WRITES
ON

All together – scale reads and writes

Master server does faster writes

Replica – with reads


Firebird master FORCED WRITES OFF
Firebird replica FORCED WRITES ON
Firebird replica

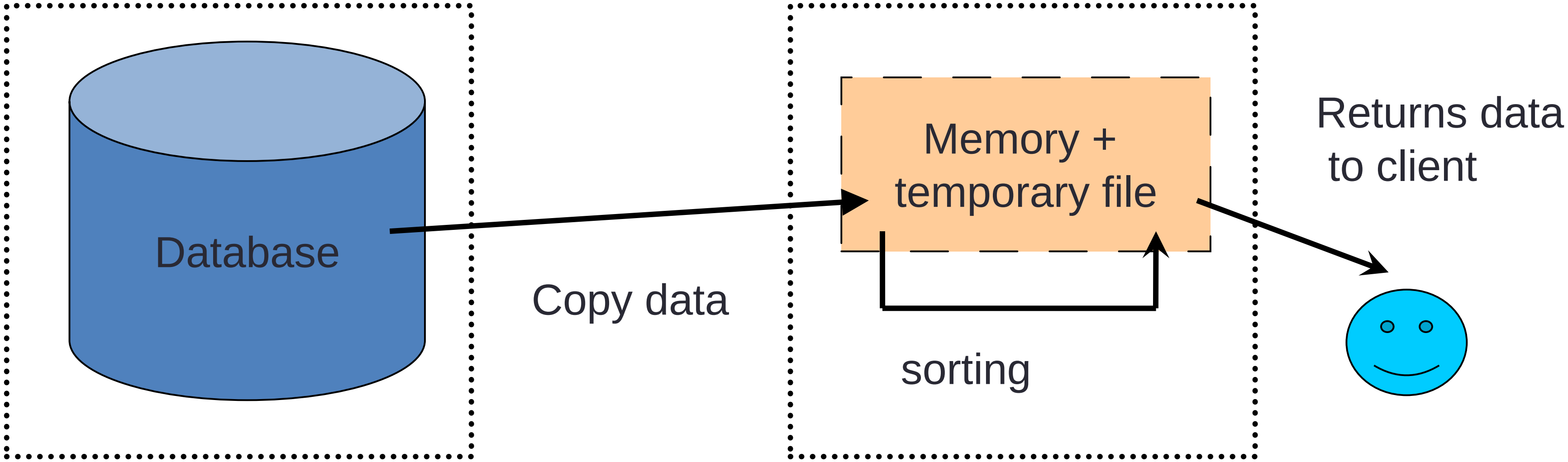
3. ENGINE PERFORMANCE IMPROVEMENTS IN HQBIRD

Engine performance improvements

- TempCacheThreshold and REFETCH
    - Diagnose and speed up large sortings
- Pool of prepared statements
    - Speed up frequent queries
- Pool of ESOE
    - Speed up cross-database queries

What is the problem with SORT?

```
select * from employee
order by first_name
PLAN SORT ((EMPLOYEE NATURAL))
```



Find big sortings: TempCacheThreshold

HQbird has new parameter of firebird.conf (2.5 and 3.0 only): Example: **TempCacheThreshold = 2000000000**

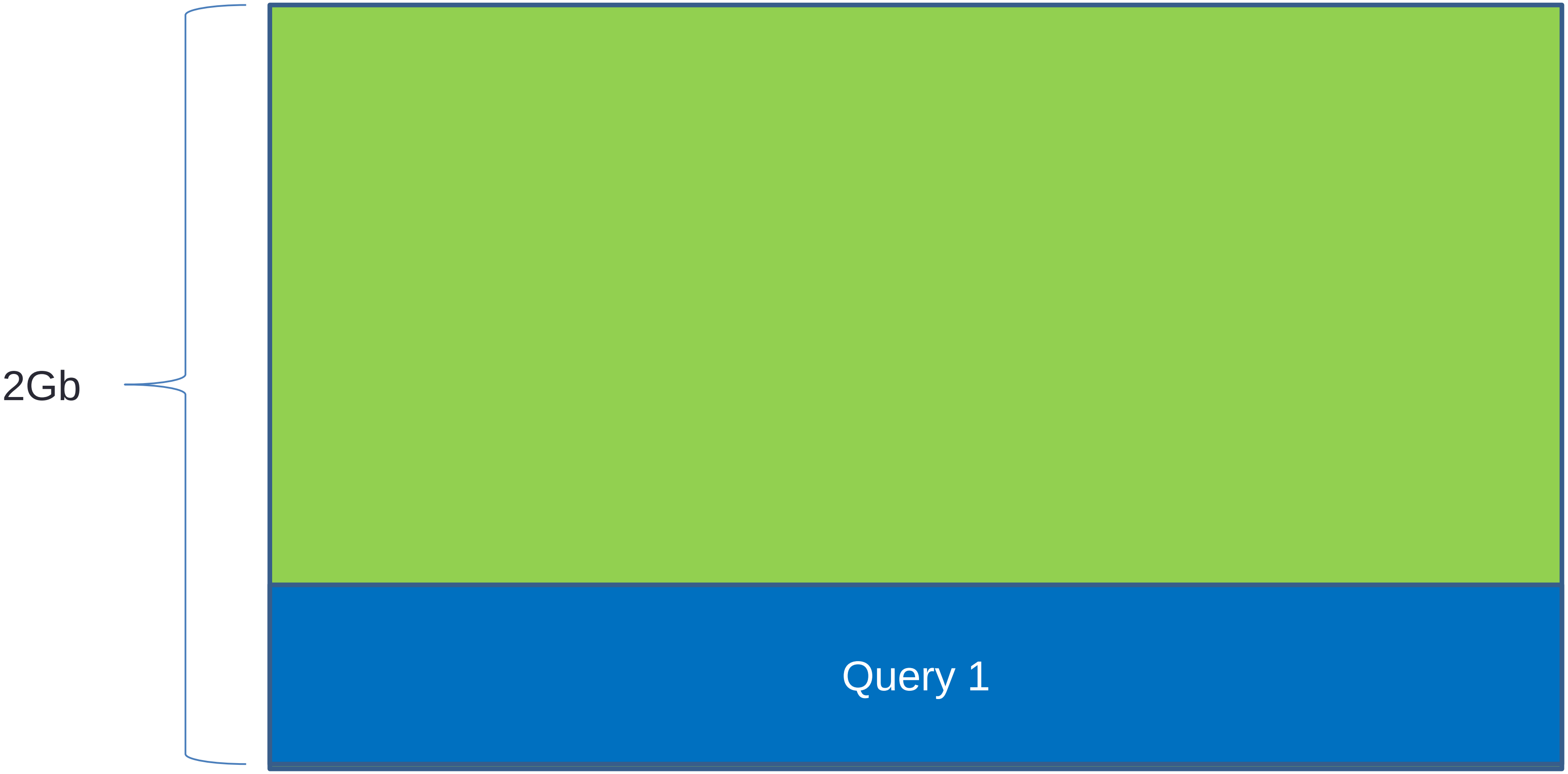
Temporary space of type "sort" has exceeded threshold of 2000000000 bytes.
Total size: 6852273600, cached: 1941962752 bytes, on disk: 4911529984 bytes.

Query:
insert into …
select distinct …

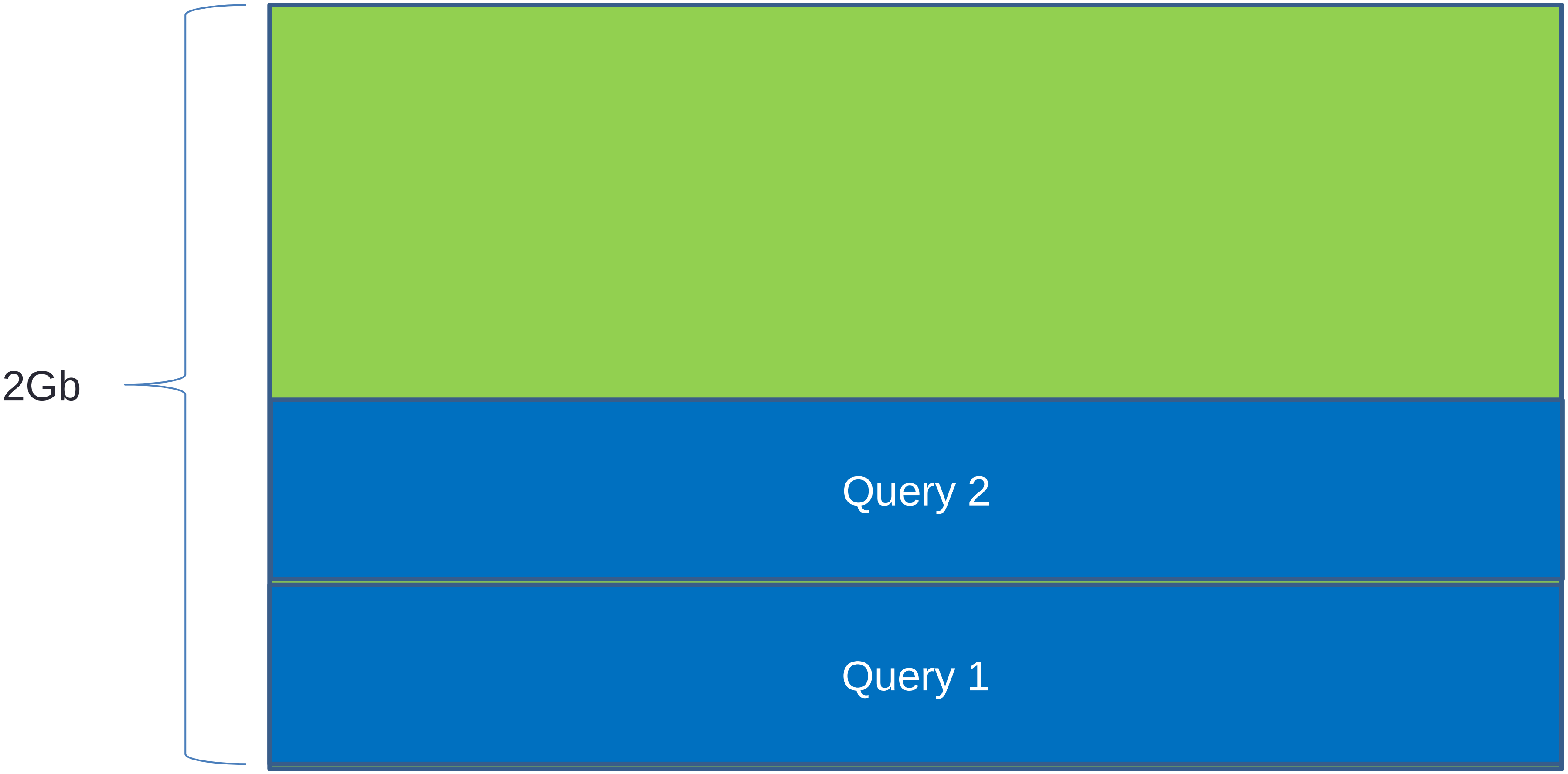
Total size – full sort size
Cached – fit in TempCacheLimit
On disk – in temporary file - in OS memory, or on disk

TempCacheLimit — allocated in RAM for sorting

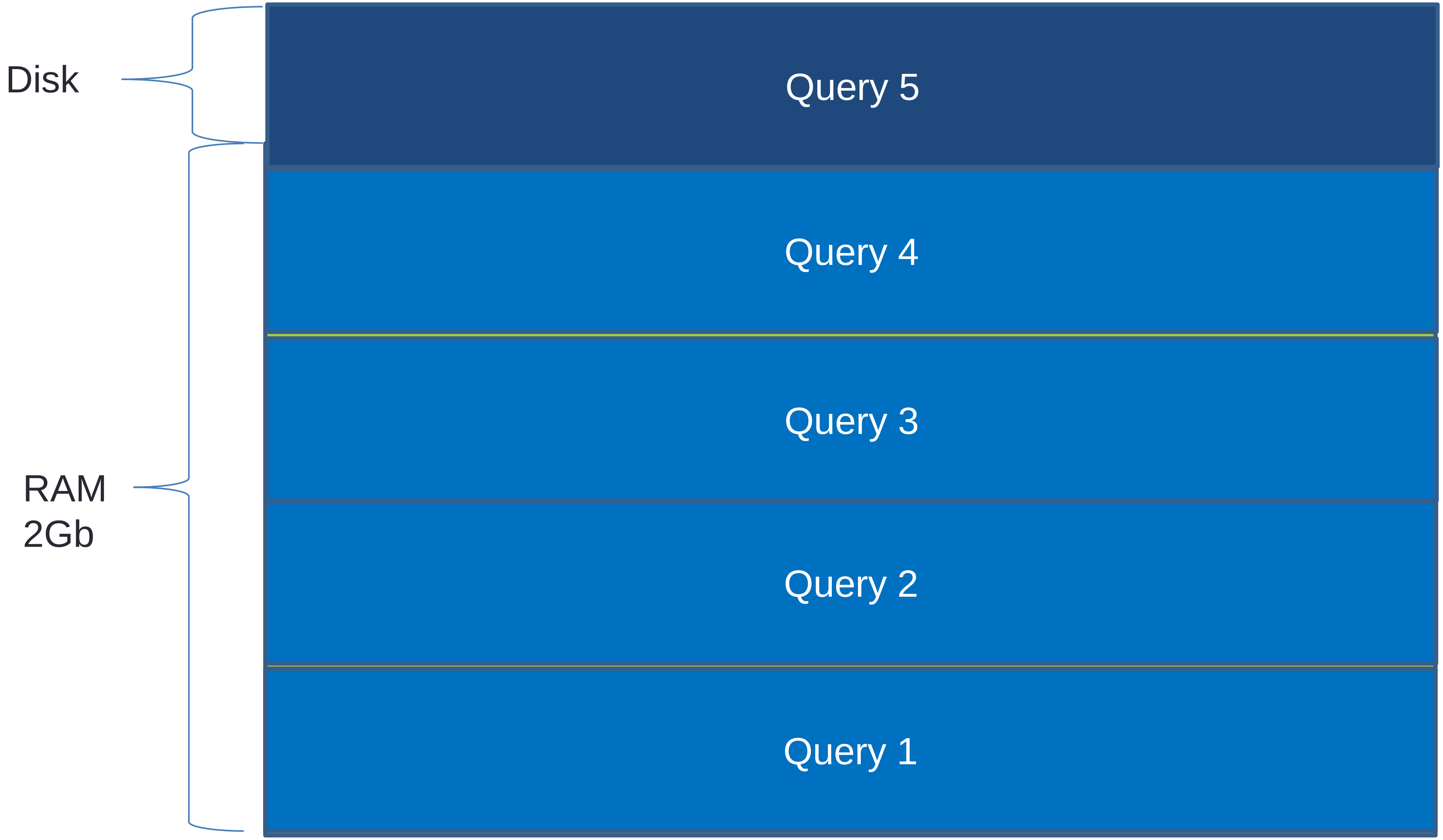
TempCacheLimit



TempCacheLimit



TempCacheLimit



Example: tables

```
Create table tmain                 ~ 50 000 records
(id integer not null primary key,
name_main varchar(50));

Create table tdetl                 ~ 50 000 records
(id integer not null primary key,
pid integer,
long_description varchar(32768));
```

Example: Query

```
select
     tdetl.name_detl
    ,tmain.name_main
    ,tdetl.long_description
from tdetl
join tmain on tdetl.pid=tmain.id
order by tdetl.name_detl
```

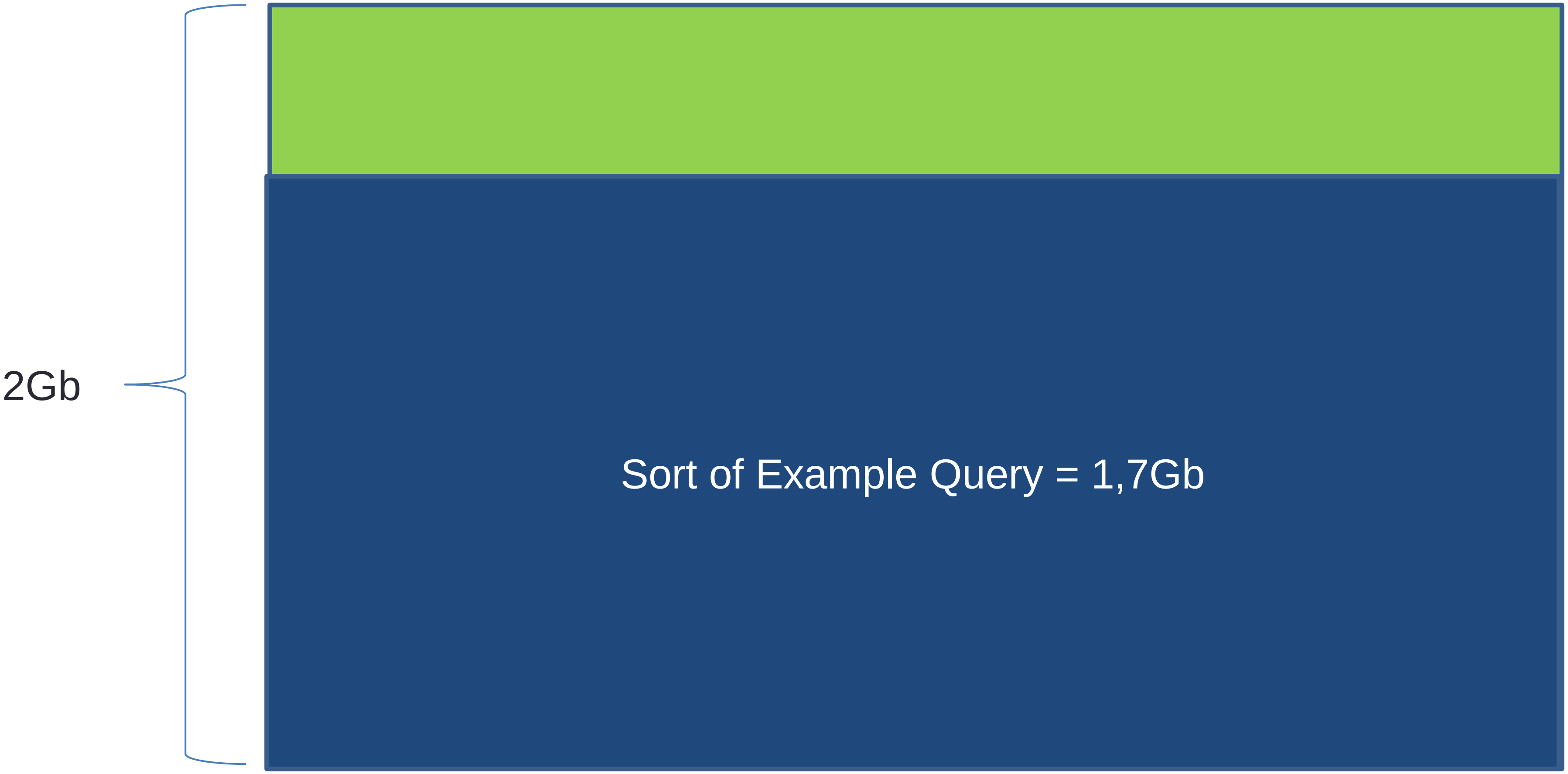
Example: plan for query

```
Select Expression
  -> Sort (record length: 32860, key length: 36)
    -> Nested Loop Join (inner)
      -> Table "TMAIN" Full Scan
        -> Filter
          -> Table "TDETL" Access By ID
            -> Bitmap
              -> Index "FK_TABLE1_1" Range Scan
(full match)
```

Required sort space ~1.7Gb

Sort (record length: 32860, key length: 36);

total size: **1 752 045 568**



Refetch instead of sort

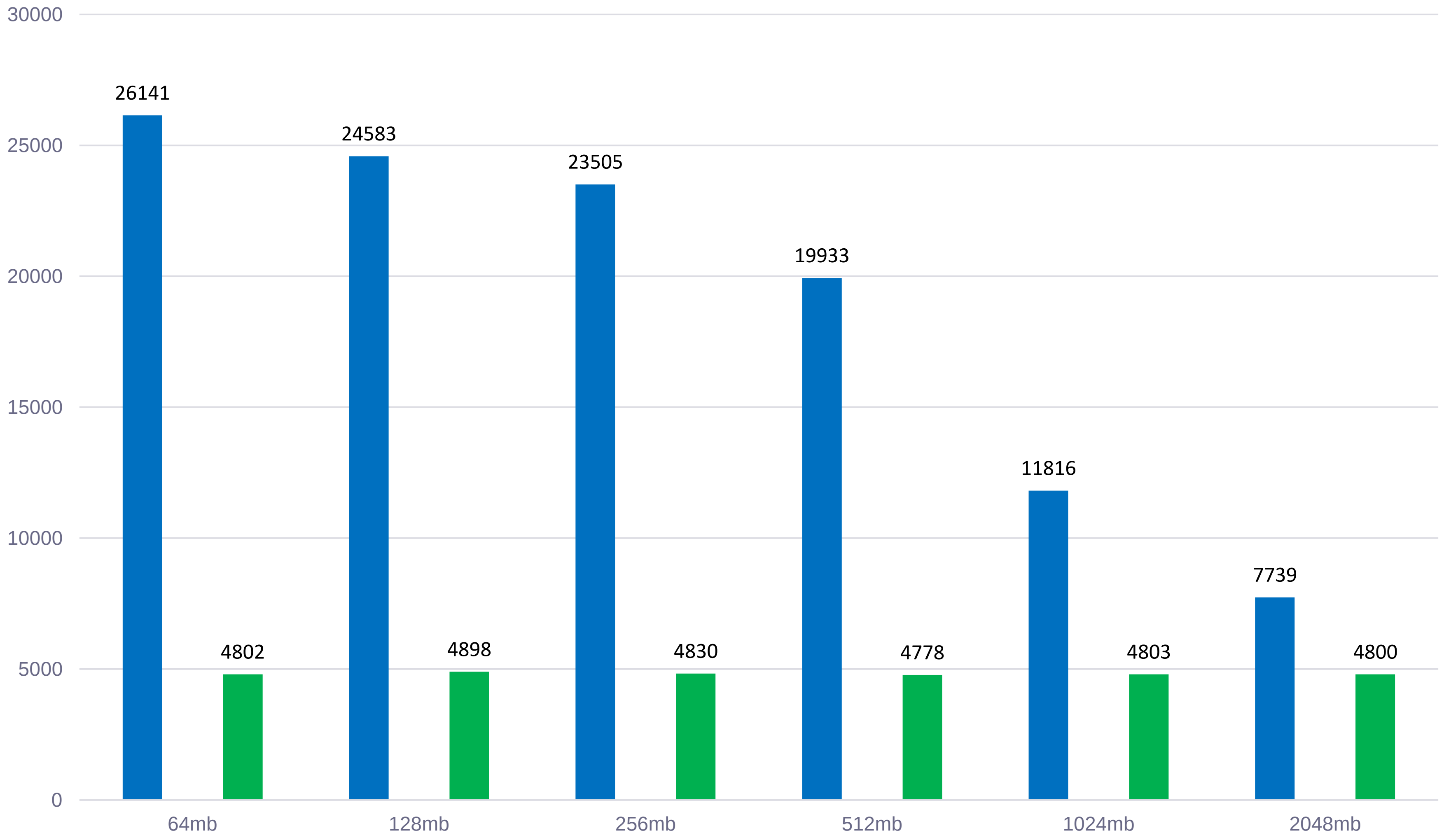
- HQbird has new parameter of firebird.conf (3.0 only): **SortDataStorageThreshold** =1024
- Restricts the size of sorting record and switch it to new plan REFETCH
  - when record length bigger than key length by SortDataStorageThreshold
- It sorts only keys! Other data are being read from database.

```
Select Expression
    -> Refetch
        -> Sort (record length: 76, key length: 36)
            -> Nested Loop Join (inner)
```

SORT vs REFETCH

Elapsed time, ms build: WI-V3.0.5.33141		
	SortDataStorageThreshold = OFF	SortDataStorageThreshold = ON (1K)
TempCacheLimit, Mb ↓	Sort (record length: 32860, key length: 36); total size: 1 752 045 568	REFETCH (record length: 76, key length: 36); total size: 4 368 000
64mb	26141	4802
128mb	24583	4898
256mb	23505	4830
512mb	19933	4778
1024mb	11816	4803
2048mb	7739	4800

SORT vs REFETCH



Summary for REFETCH

- Great choice for reports and queries with many fields (more than 4096 bytes): they are sorted 1.5-5 times faster
- Reduced requirements for TempCacheLimit and improves overall performance
- Available in HQbird Enterprise 2020

Pool of prepared statements

- New parameter in firebird.conf (only 3.0)
  - DSQLCacheSize = 8 (recommended 8-16)
- Allows to cache prepared statements in each connection
  - Increase memory consumption by 1-10Mb per connection
- Speed up frequent non-prepared queries up to 6-7 times, from 40ms to 5-7ms

4. SECURITY, AUTHENTICATION, ENCRYPTION IN HQBIRD 2020

Security, Authentication, Encryption

- Security: RSA-UDR
  - sign documents with electronic signature
- Authentication: plugin libCluster
  - perform Execute Statement On External without passwords
- Encryption
  - Encrypt your databases and backups with AES256

RSA – UDR to sign documents and sources

- To protect documents from modification:
  - Sources of your DDL,
  - Documents in VARCHARS and BLOBs

Example of RSA-UDR usage

```
C:\HQbird\Firebird30>isql localhost:c:\temp\rsatest.fdb
-user SYSDBA -pass masterkey
Database: localhost:c:\temp\rsatest.fdb, User: SYSDBA
SQL> show functions;
Global functions

Function Name                     Invalid Dependency,
Type
================================= =======
=====================================
RSA_PRIVATE_KEY
RSA_PUBLIC_KEY
RSA_SIGN
RSA_VERIFY
SHA256
```

Example of libRSA usage

-
  ```
  SQL> show tables;
         TBL
  SQL> show table TBL;
  DOC                                BLOB segment 80,
  subtype BINARY Nullable
  DIGEST                             VARCHAR(32)
  CHARACTER SET OCTETS Nullable
  SALTLEN                            INTEGER Nullable
  PRIVATE_KEY                        VARCHAR(2048)
  CHARACTER SET OCTETS Nullable
  SIGN                               VARCHAR(1024)
  CHARACTER SET OCTETS Nullable
  PUBLIC_KEY                         VARCHAR(512)
  CHARACTER SET OCTETS Nullable
  BAD_SIGN                           VARCHAR(1024)
  CHARACTER SET OCTETS Nullable
  ```

Example of libRSA usage

```
---clean the test table
SQL>delete from tbl
SQL>commit;

--- generate private key and write
--- it into table TBL
SQL>insert into tbl(PRIVATE_KEY)
values(rsa_private_key(1024));

---  generate public key
SQL>update tbl set PUBLIC_KEY =
rsa_public_key(PRIVATE_KEY);
```

Example of libRSA usage

```
---- create BLOB document
SQL>update TBL set DOC='testtesttest';

--- and calculate its digest
SQL>update tbl set digest = sha256(doc);
```

Example of libRSA usage

-
  ```
  ---sign document and remember its signature
  SQL>update tbl set sign =
  rsa_sign(digest, PRIVATE_KEY, 8);

  --- check the signature
  ```

-
  ```
  SQL> select RSA_VERIFY(SIGN, DIGEST,
  PUBLIC_KEY, SALTLEN) from tbl;

  RSA_VERIFY
  ==========
  <true>
  ```

Example of libRSA usage

```
---change the document (BLOB)
SQL> update TBL set DOC='testtesttest222';

----recalculate its digest
SQL> update tbl set digest = sha256(doc);

--- check signature
SQL> select rsa_verify(sign, digest,
PUBLIC_KEY, 8) from tbl;

RSA_VERIFY
==========
<false>
```

RSA library extra features

```
function hex2bin
function bin2hex

SQL> set list;
SQL> select bin2hex('Test string') from
rdb$database;

BIN2HEX                          5465737420737472696E6
7

SQL> select cast
(hex2bin('5465737420737472696E67') as
varchar(32)) from rdb$database;

CAST                             Test string
```

libCluster – authentication plugin

- ESOE
- EXECUTE STATEMENT “SELECT..” ON EXTERNAL AS USER “MYUSER” PASSWORD “mypassword”

- libCluster establishes trust and mappings between different Firebird servers and users, so it is possible to run without password:

- EXECUTE STATEMENT “SELECT..” ON EXTERNAL AS USER “MYUSER”;

Encryption plugin

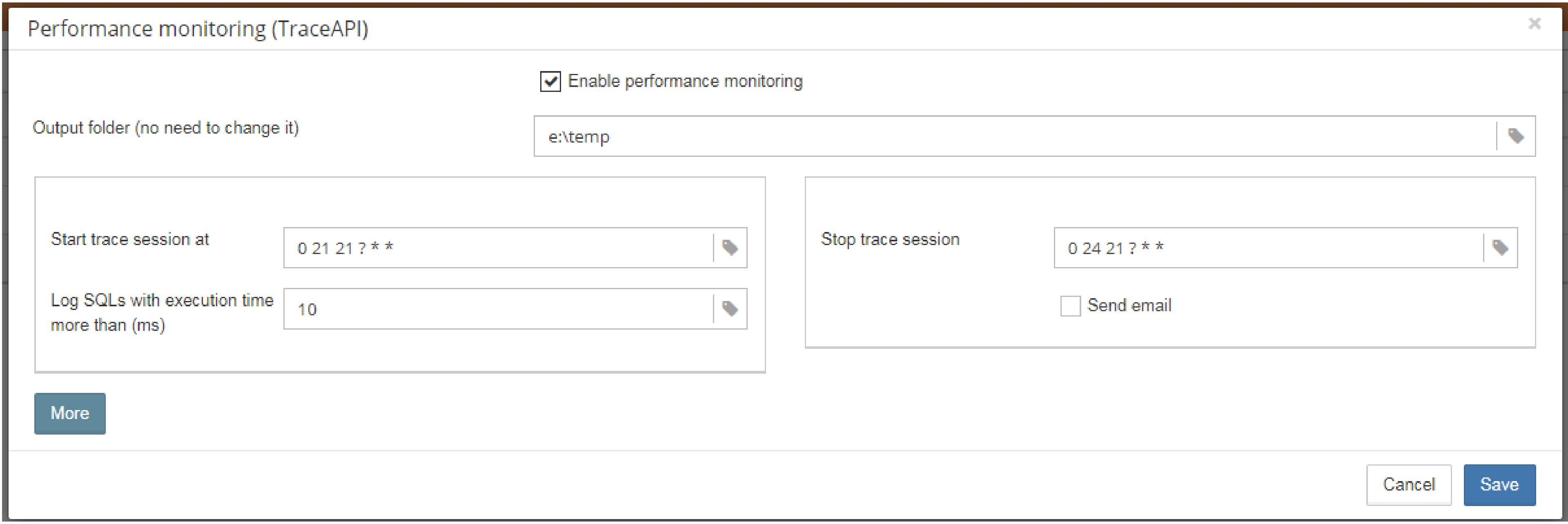
- HQbird 2019 includes 1 server license of IBSurgeon Encryption Plugin
  - ISV Subscription includes unlimited plugin (without sources)
- AES256 Strong Encryption
- Examples for Delphi, .NET, .Java, PHP
- Includes special tools:
  - Gbak with encryption support
  - Decrypt tool for low-level decryption
  - FirebirdSQLStudio has support for encrypted databases

5. PERFORMANNCE MONITORING AND REPORTS HQBIRD 2020

New performance report

- Time-Summary
  - The most time consuming queries – they should be optimized first!
- Fetches
  - The most CPU-consuming queries
- Reads and Writes
  - The most disk-intensive queries
- Process Summary
  - All key parameters are grouped by applications

Easy to setup



Performance monitoring (TraceAPI)

☑ Enable performance monitoring

Output folder (no need to change it): e:\temp

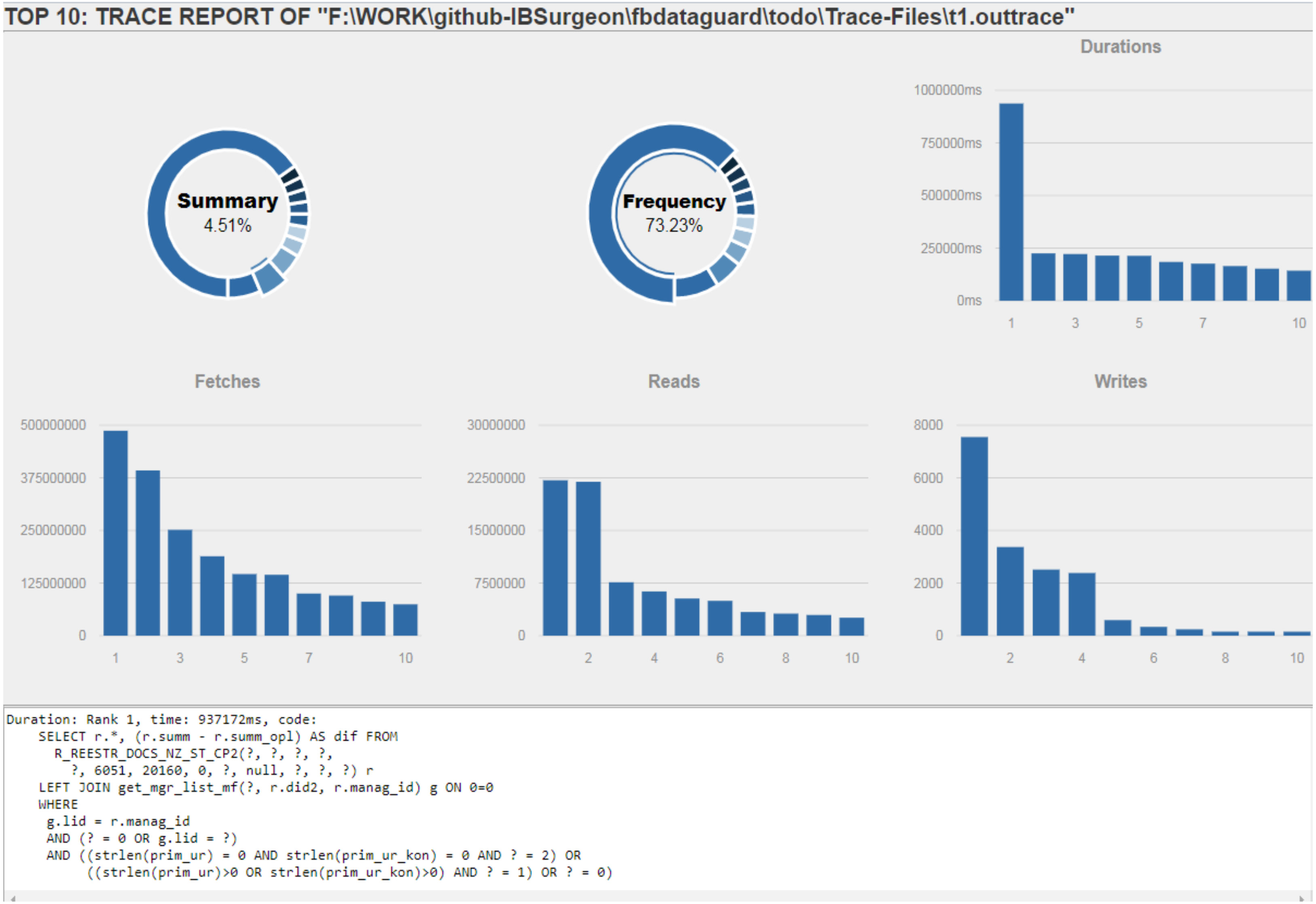
Start trace session at: 0 21 21 ? * *

Log SQLs with execution time more than (ms): 10

Stop trace session: 0 24 21 ? * *

☐ Send email

Demo performance reports

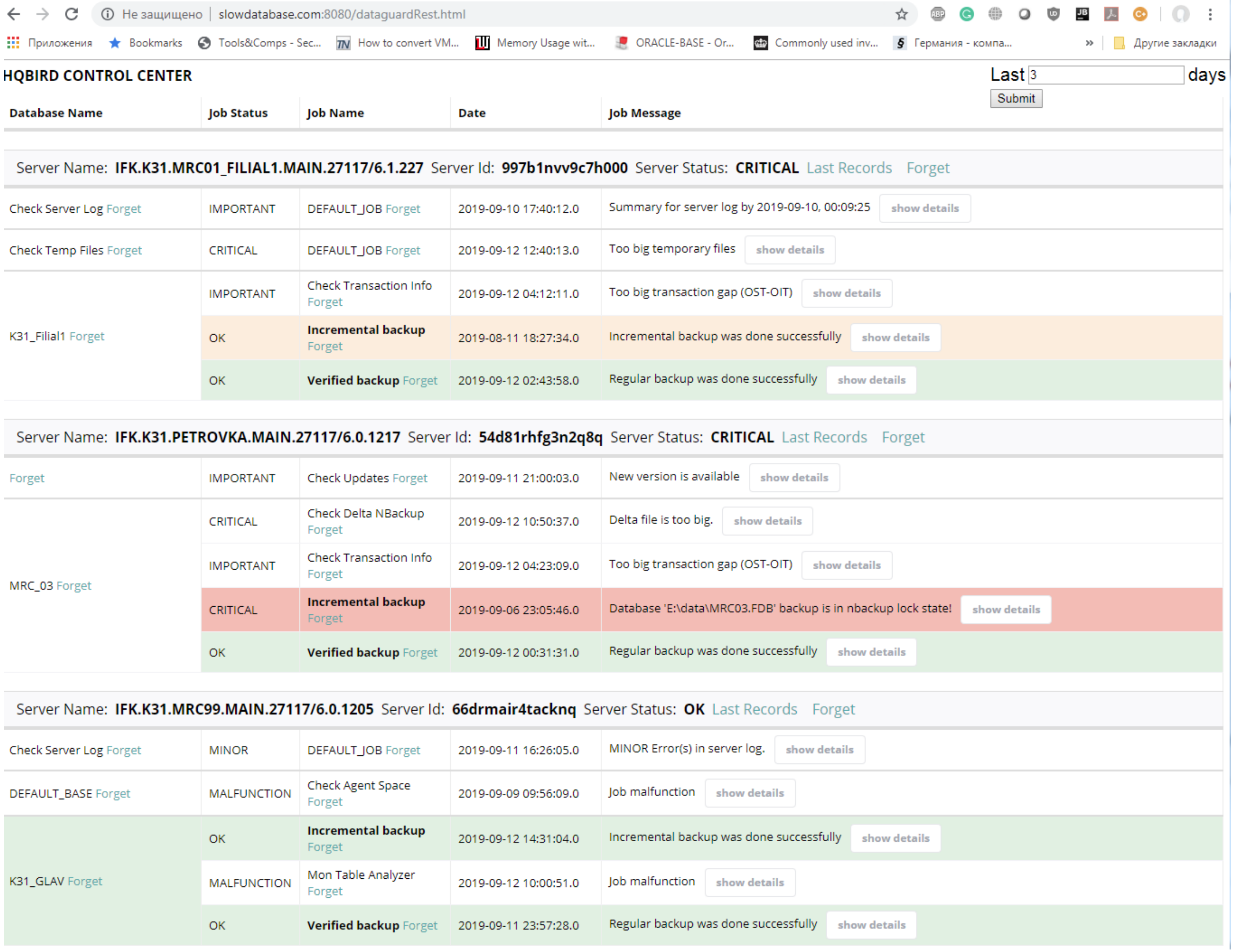


TOP 10: TRACE REPORT OF "F:\WORK\github-IBSurgeon\fbdataguard\todo\Trace-Files\t1.outtrace"

```
Duration: Rank 1, time: 937172ms, code:
    SELECT r.*, (r.summ - r.summ_opl) AS dif FROM
      R_REESTR_DOCS_NZ_ST_CP2(?, ?, ?, ?,
        ?, 6051, 20160, 0, ?, null, ?, ?, ?) r
    LEFT JOIN get_mgr_list_mf(?, r.did2, r.manag_id) g ON 0=0
    WHERE
     g.lid = r.manag_id
     AND (? = 0 OR g.lid = ?)
     AND ((strlen(prim_ur) = 0 AND strlen(prim_ur_kon) = 0 AND ? = 2) OR
          ((strlen(prim_ur)>0 OR strlen(prim_ur_kon)>0) AND ? = 1) OR ? = 0)
```

5. CONTROL CENTER FOR HQBIRD

Demo Control Center



HQBIRD CONTROL CENTER

Last 3 days

Submit

Database Name	Job Status	Job Name	Date	Job Message	
Server Name: **IFK.K31.MRC01_FILIAL1.MAIN.27117/6.1.227** Server Id: **997b1nvv9c7h000** Server Status: **CRITICAL** Last Records Forget					
Check Server Log Forget	IMPORTANT	DEFAULT_JOB Forget	2019-09-10 17:40:12.0	Summary for server log by 2019-09-10, 00:09:25	show details
Check Temp Files Forget	CRITICAL	DEFAULT_JOB Forget	2019-09-12 12:40:13.0	Too big temporary files	show details
K31_Filial1 Forget	IMPORTANT	Check Transaction Info Forget	2019-09-12 04:12:11.0	Too big transaction gap (OST-OIT)	show details
	OK	**Incremental backup** Forget	2019-08-11 18:27:34.0	Incremental backup was done successfully	show details
	OK	**Verified backup** Forget	2019-09-12 02:43:58.0	Regular backup was done successfully	show details
Server Name: **IFK.K31.PETROVKA.MAIN.27117/6.0.1217** Server Id: **54d81rhfg3n2q8q** Server Status: **CRITICAL** Last Records Forget					
Forget	IMPORTANT	Check Updates Forget	2019-09-11 21:00:03.0	New version is available	show details
MRC_03 Forget	CRITICAL	Check Delta NBackup Forget	2019-09-12 10:50:37.0	Delta file is too big.	show details
	IMPORTANT	Check Transaction Info Forget	2019-09-12 04:23:09.0	Too big transaction gap (OST-OIT)	show details
	CRITICAL	**Incremental backup** Forget	2019-09-06 23:05:46.0	Database 'E:\data\MRC03.FDB' backup is in nbackup lock state!	show details
	OK	**Verified backup** Forget	2019-09-12 00:31:31.0	Regular backup was done successfully	show details
Server Name: **IFK.K31.MRC99.MAIN.27117/6.0.1205** Server Id: **66drmair4tacknq** Server Status: **OK** Last Records Forget					
Check Server Log Forget	MINOR	DEFAULT_JOB Forget	2019-09-11 16:26:05.0	MINOR Error(s) in server log.	show details
DEFAULT_BASE Forget	MALFUNCTION	Check Agent Space Forget	2019-09-09 09:56:09.0	Job malfunction	show details
K31_GLAV Forget	OK	**Incremental backup** Forget	2019-09-12 14:31:04.0	Incremental backup was done successfully	show details
	MALFUNCTION	Mon Table Analyzer Forget	2019-09-12 10:00:51.0	Job malfunction	show details
	OK	**Verified backup** Forget	2019-09-11 23:57:28.0	Regular backup was done successfully	show details

About IBSurgeon

IBSurgeon

- Tools and consulting
- Sponsor of Firebird Foundation
- Founded in 2002: 17 years of Firebird and InterBase recoveries and consulting
- Based in Moscow, Russia

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

Thank you!

- support@ib-aid.com