database/sql: плохой, хороший, злой.

Опыт разработки драйвера для распределённой СУБД YDB

Алексей Мясников,
Яндекс, YDB

Golang Conf 2023

Алексей Мясников

- Руководитель команды в YDB
- К. т. н.
- Писал код на 20+ языках программирования
- Ментор на курсе Go-разработчик
- На Go пишу за деньги с 2018 года
- С 2021 года работаю в YDB
- Разобрал за 2 месяца ~500 багов

Эволюция пакета database/sql в Go

Буду рассказывать

1

Основные качественные изменения стандартной библиотеки Go

2

Проекция изменений стандартной библиотеки Go на работу с БД

3

Проекция изменений стандартной библиотеки Go на работу с БД YDB

Что такое YDB?

Распределённая опенсорс-платформа данных

t.me/ydb_ru

t.me/ydb_en

github.com

ydb.tech

- SQL для OLTP
- SQL для OLAP
- Горизонтальное масштабирование
- Транзакции с гарантиями ACID
  в нескольких AZ
- Работоспособность и автоматическое
  восстановление при отказах
- Масштабирование на миллионы транзакций
  в секунду и сотни терабайт данных
- Координация распределённых систем
  (like ZooKeeper)
- Доставка сообщений (like Kafka)
- Безостановочная работа 24/7/365

Рандомные факты о YDB…

Внутри Яндекса, Yandex Cloud, внешние облака, on-premise

Самый большой кластер ~10 000 узлов

Яндекс Метрика

- 1,5 М транзакций в секунду
- 1,5 петабайта данных
- Одна из самых больших таблиц

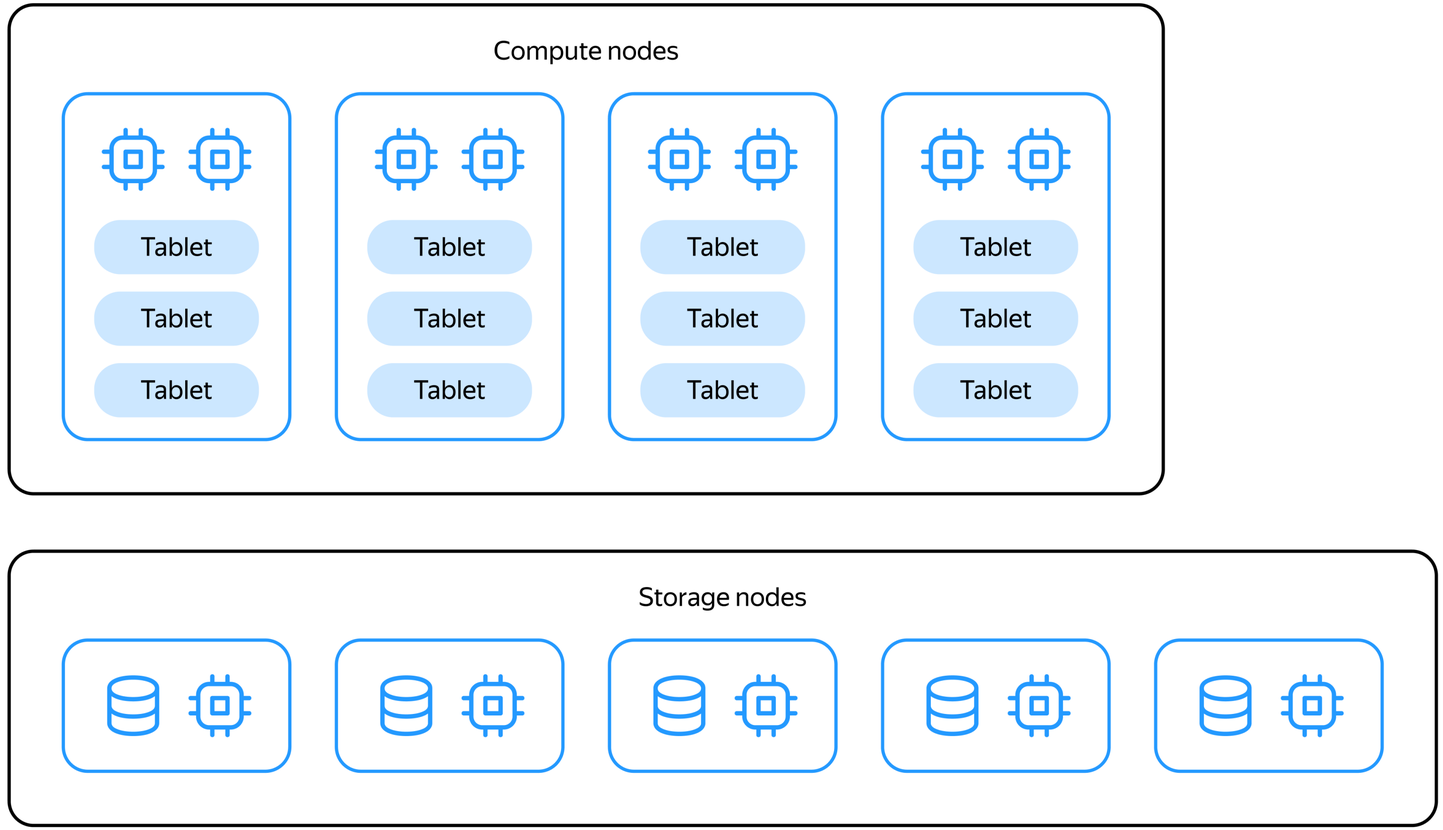
  35 000 шардов

  370 миллиардов строк

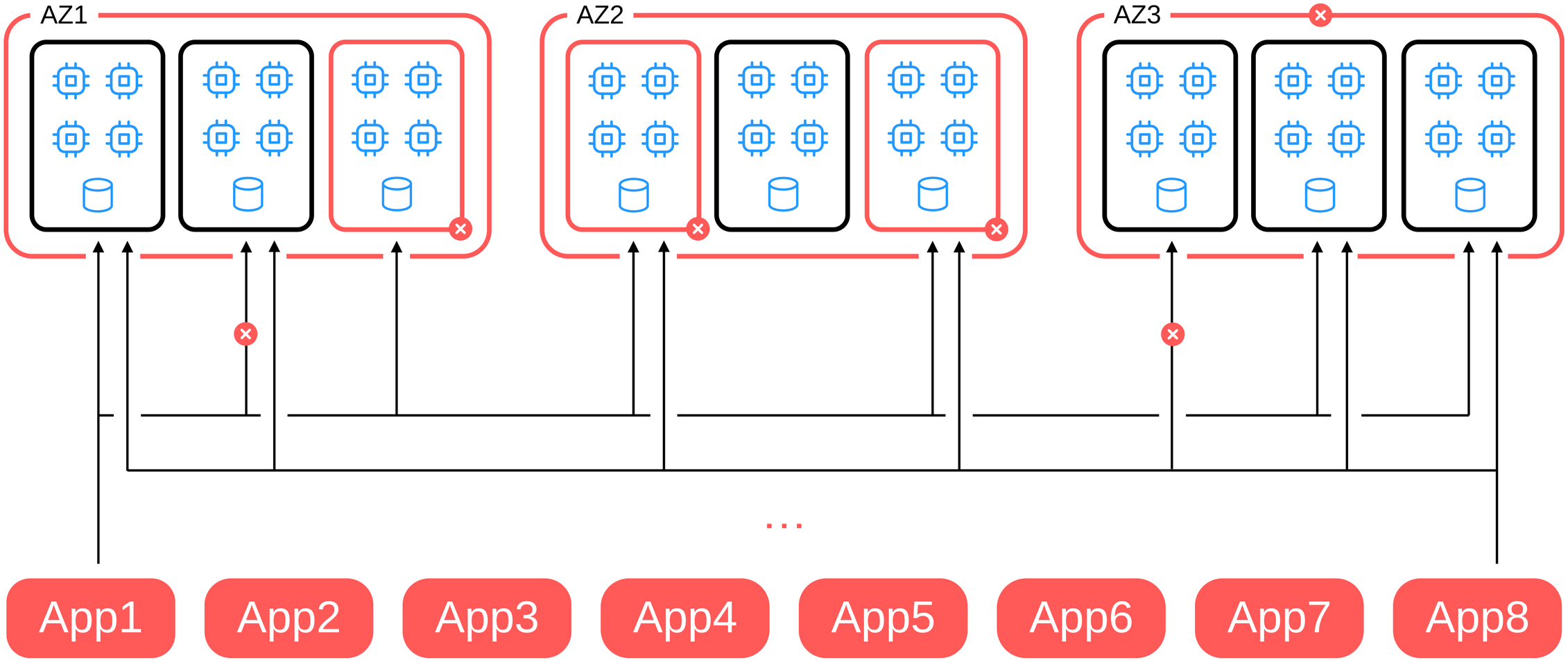
  66 терабайт данных

Разделение слоёв Compute и Storage

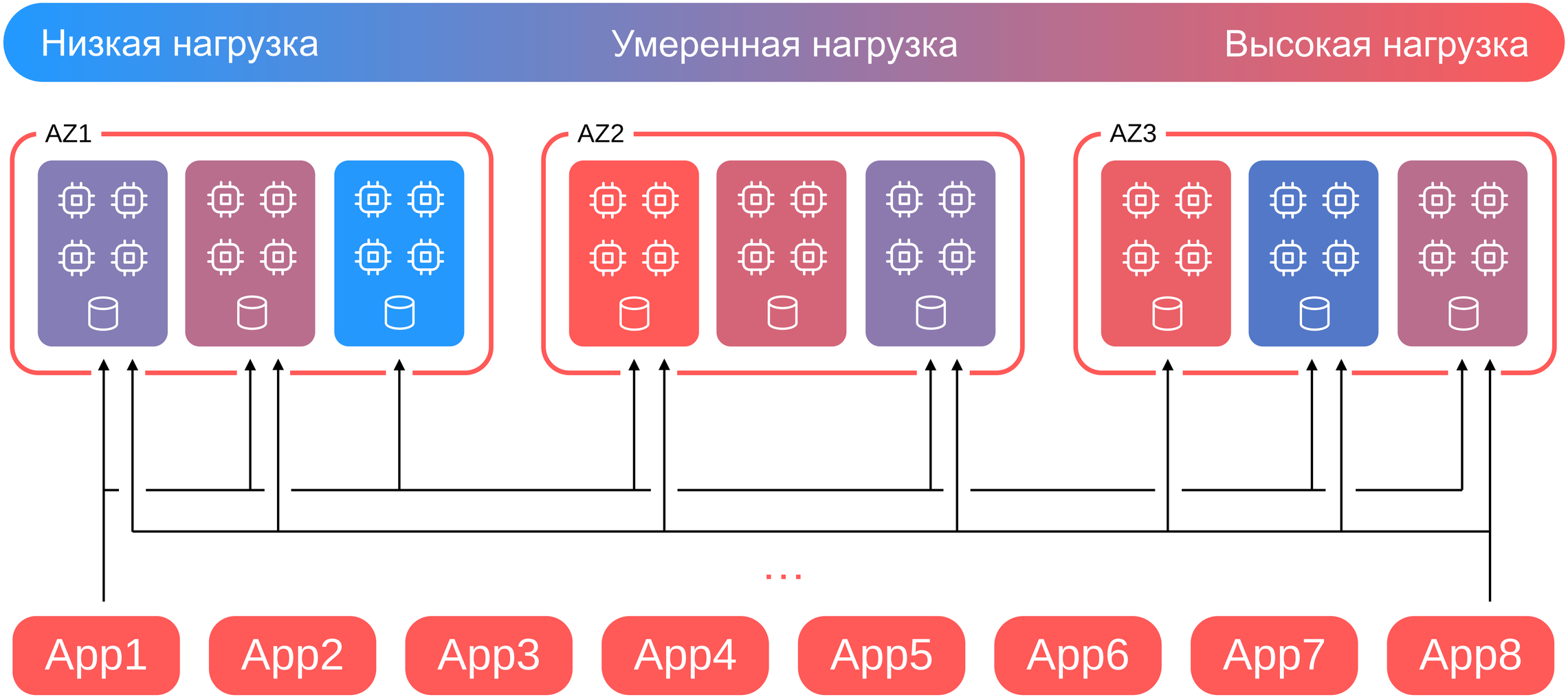
- Среды выполнения для таблеток и запросов запущены на вычислительных узлах
- Данные размещены на узлах хранения
- Каждый «узел» — процесс в операционной системе



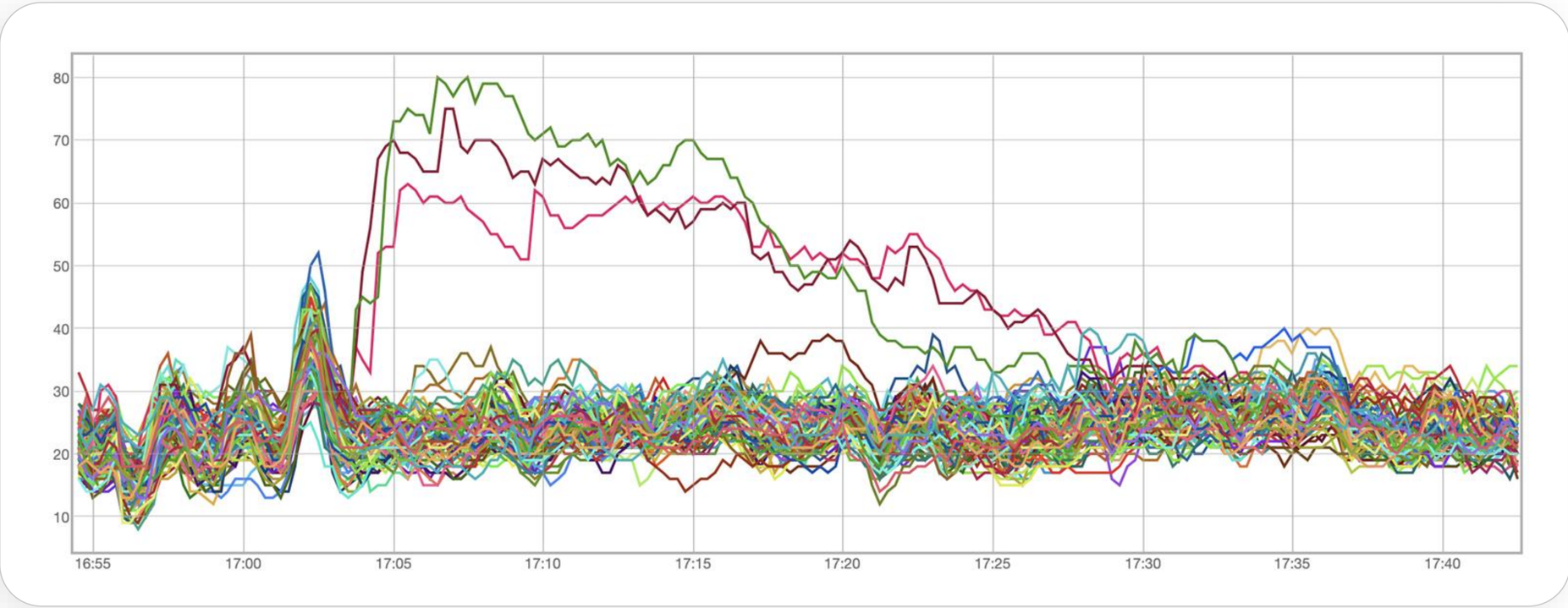
YDB с точки зрения клиентского приложения



Клиентская балансировка

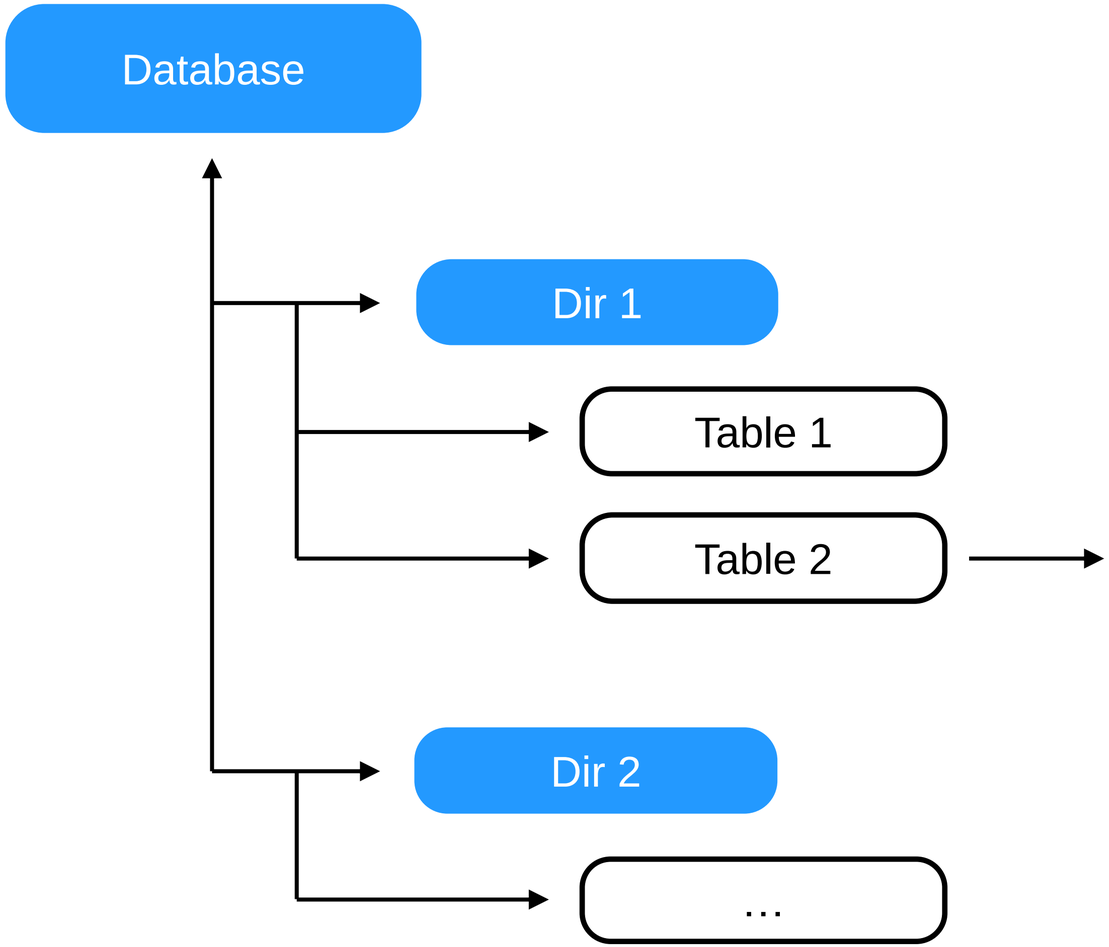


Серверная балансировка



- Упрощаем клиентскую балансировку
- Обеспечиваем равномерную нагрузку на ноды
- Аккуратно закрываем сессии

Таблицы организованы в иерархию



Transactional DB

- Access time
- Size

- Columns
- Primary key
- Secondary Index
- Shards
- Statistics
- Configuration

YQL

- Диалект SQL
- Строгая типизация
- Именованные подзапросы
- Явная параметризация
- Богатый набор встроенных функций

YQL

1

Стандартные DQL-конструкции

SELECT

JOIN

GROUP BY

ORDER BY

LIMIT

2

Стандартные DML-конструкции

INSERT, UPDATE, DELETE

UPSERT / REPLACE

UPDATE ON

DELETE ON

3

Стандартные DDL-конструкции

CREATE TABLE

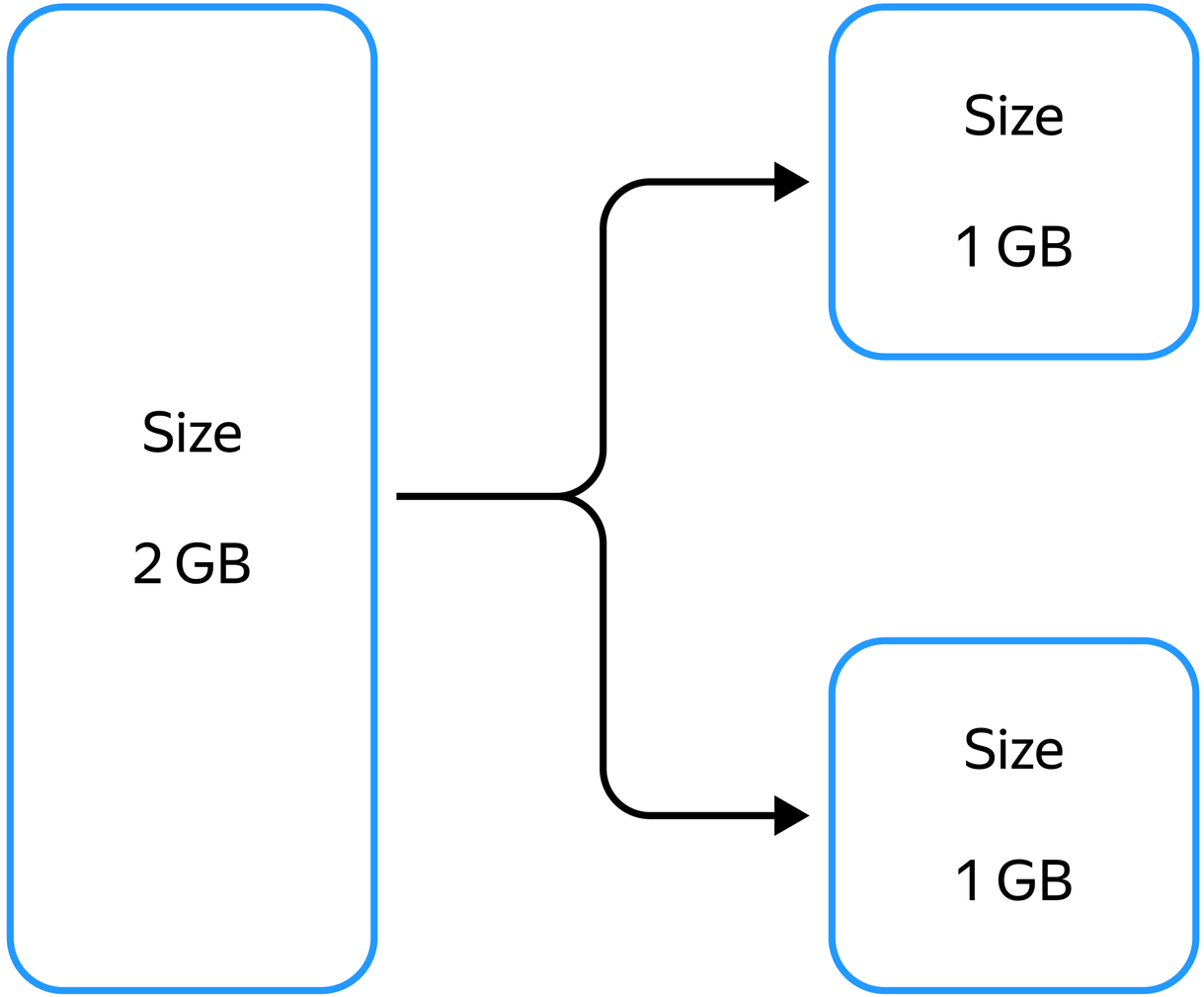
DROP TABLE

ALTER TABLE

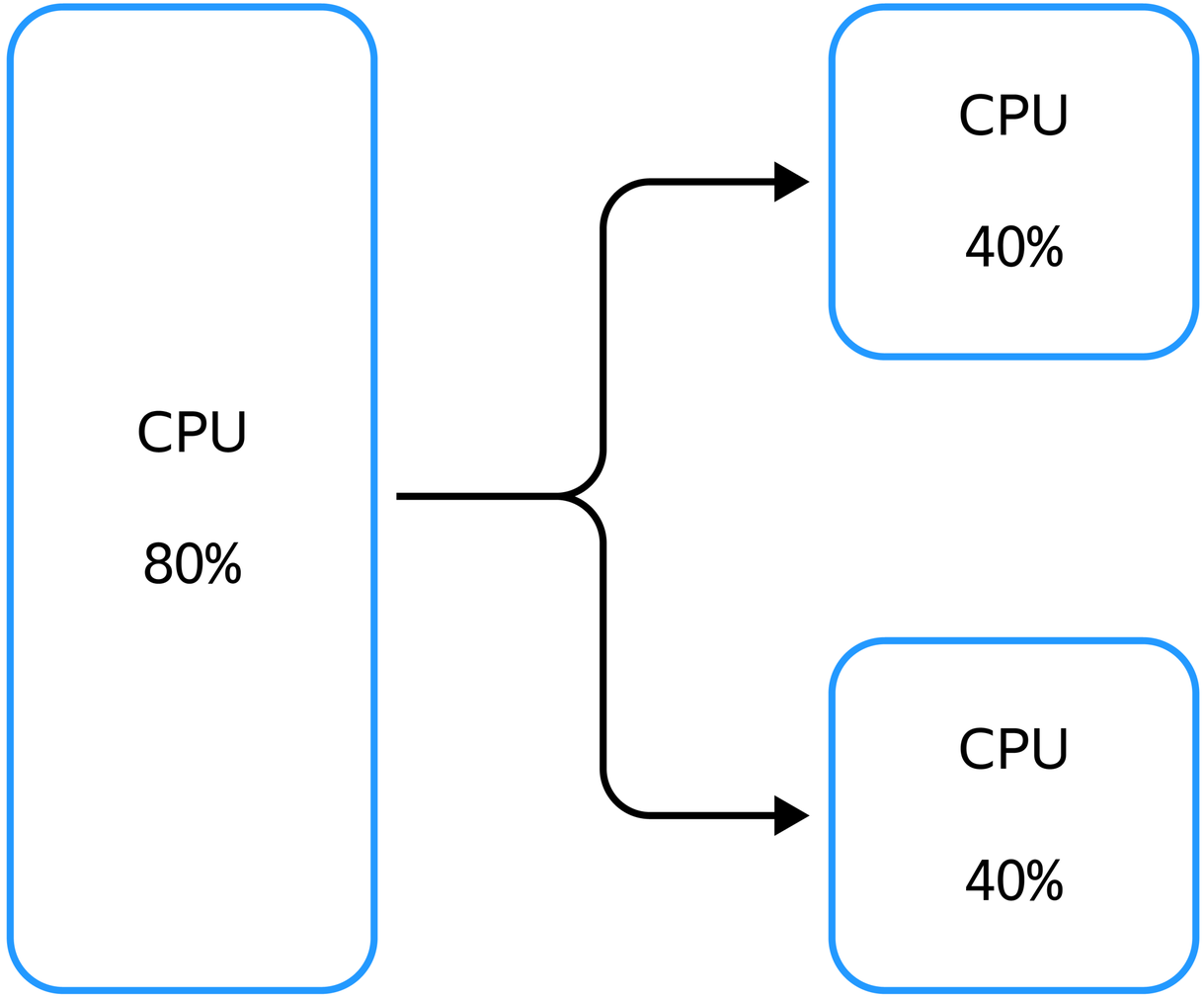
Данные упорядочены по первичному ключу


```
CREATE TABLE account_history (
    account_id Text NOT NULL,
    operation_id Text NOT NULL,
    name Text NOT NULL,
    amount Int64 NOT NULL,
    PRIMARY KEY (account_id, operation_id)
)
```

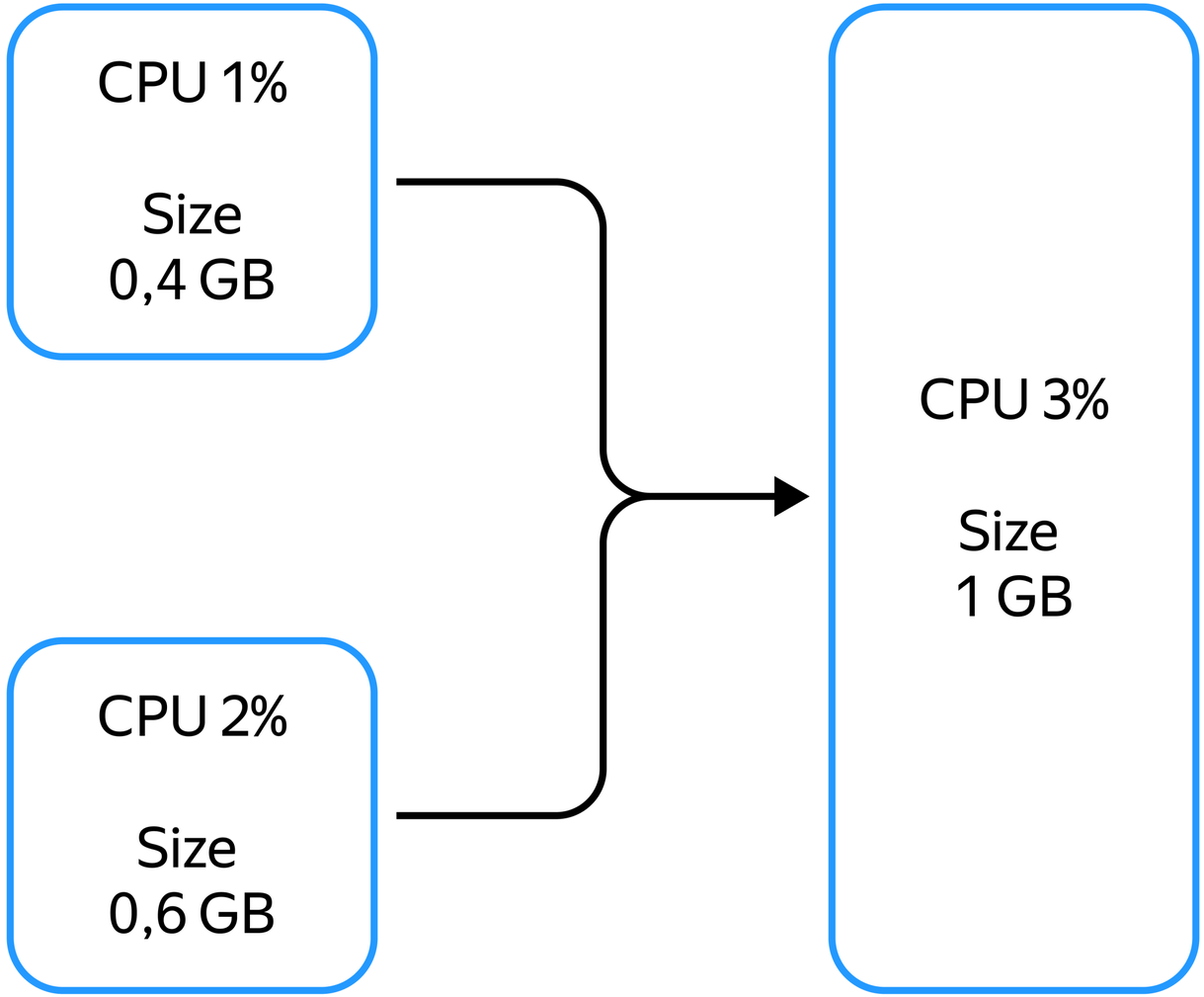
Шардирование таблиц по объёму



Шардирование таблиц по нагрузке



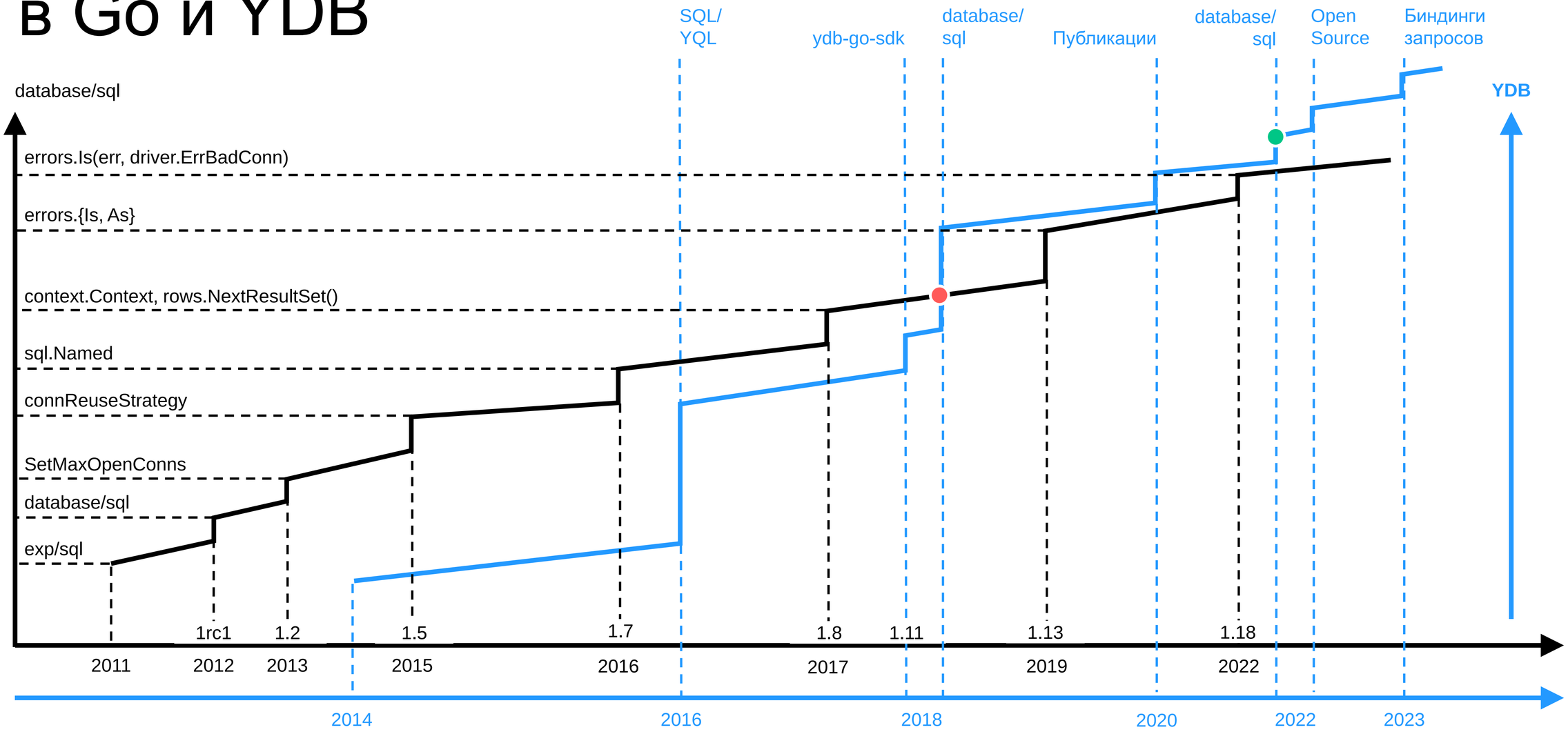
Автомёрдж таблиц после шардирования



Ручное шардирование


```
CREATE TABLE account_history (
    …
    PRIMARY KEY (account_id, operation_id)
) WITH (
    AUTO_PARTITIONING_MIN_PARTITIONS_COUNT=5,
    PARTITION_AT_KEYSPARTITION_AT_KEYS=('2', '4', '6', '8')
)
```

Эволюция пакета database/sql в Go и YDB



Что было в мире для работы с БД?

 — ODBC 3.8

ODBC

Что было в мире для работы с БД?

JDBC 4.0

```
try (PreparedStatement pstmt = conn.prepareStatement(
 "UPDATE employees SET position=? WHERE emp_id=?")) {
 pstmt.setString(1, "lead developer");
 pstmt.setInt(2, 1);
 pstmt.executeUpdate();
}
```

Что было в мире для работы с БД?

JDBC 4.0

```
Connection connection = dataSource.getConnection();
try (connection) {
  connection.setAutoCommit(false);
  PreparedStatement pstmt = conn.prepareStatement(
     "UPDATE employees SET position=? WHERE emp_id=?")) {
  pstmt.setString(1, "lead developer");
  pstmt.setInt(2, 1);
  pstmt.executeUpdate();
  connection.commit();
} catch (SQLException e) {
  connection.rollback();
}
```

exp/sql

```
type Driver interface {
    Open(name string) (Conn, error)
}
type Conn interface {
    Prepare(query string) (Stmt, error)
    Close() error
    Begin() (Tx, error)
}
type Stmt interface {
    Close() error
    NumInput() int
    Exec(args []Value) (Result, error)
    Query(args []Value) (Rows, error)
}
type Tx interface {
    Commit() error
    Rollback() error
}
```

30.09.2011 357f2cb1
Brad Fitzpatrick
codereview.appspot.com/4973055

Точно SQL?


```
db, err := sql.Open("sql-storage", "")
if err != nil { /*fallback*/ }
_, err = db.Exec("SET key1=$1", 1)
if err != nil { /*fallback*/ }
rows, err := db.Query("GET key1")
if err != nil { /*fallback*/ }
var value int
if !rows.Next() { panic("no rows") }
if err = rows.Scan(&value); err != nil { /*fallback*/ }
```

driver.Result


```
// Result is the result of a query execution.
type Result interface {
  // LastInsertId returns the database's auto-generated ID
  // after, for example, an INSERT into a table with primary
  // key.
  LastInsertId() (int64, error)

  // RowsAffected returns the number of rows affected by the
  // query.
  RowsAffected() (int64, error)
}
```

Казалось бы, всё просто, но есть еще неявные интерфейсы в database/sql/driver


```
type DriverContext interface {...}
type Connector interface {...}
type Pinger interface {...}
type Execer interface {...}
type ExecerContext interface {...}
type Queryer interface {...}
type QueryerContext interface {...}
type ConnPrepareContext interface {...}
type ConnBeginTx interface {...}
type SessionResetter interface {...}
...
```

Казалось бы, всё просто, но есть еще неявные интерфейсы в database/sql/driver


```
...
type Validator interface {...}
type StmtExecContext interface {...}
type StmtQueryContext interface {...}
type NamedValueChecker interface {...}
type ColumnConverter interface {...}
type RowsNextResultSet interface {...}
type RowsColumnTypeScanType interface {...}
type RowsColumnTypeDatabaseTypeName interface {...}
type RowsColumnTypeLength interface {...}
type RowsColumnTypeNullable interface {...}
type RowsColumnTypePrecisionScale interface {...}
```

Как это работает?


```
if driverCtx, ok := driveri.(driver.DriverContext); ok {
  connector, err := driverCtx.OpenConnector(dataSourceName)
  if err != nil {
     return nil, err
  }
  return OpenDB(connector), nil
}
```

Prepare

- Эффективность
- Безопасность

Prepare — обман


```
_, err = db.Exec("INSERT INTO tbl (id, value) VALUES ($1, $2);", 1, 2)
```

Prepare — обман

github.com/jackc/pgx

Prepare — обман

github.com/jackc/pgx

clck.ru/36ePVs

```
func (p *Pipeline) SendPrepare(
    name, sql string, paramOIDs []uint32,
) {
    if p.closed {
        return
    }
    p.pendingSync = true
    p.conn.frontend.SendParse(&pgproto3.Parse{Name: name, Query: sql, ParameterOIDs: paramOIDs})
    p.conn.frontend.SendDescribe(&pgproto3.Describe{ObjectType: 'S', Name: name})
}
```

Prepare — обман

github.com/jackc/pgx

clck.ru/36ePuH

```
sd, err = c.pgConn.Prepare(ctx, psName, sql, nil)
if err != nil {
    return nil, err
}
```

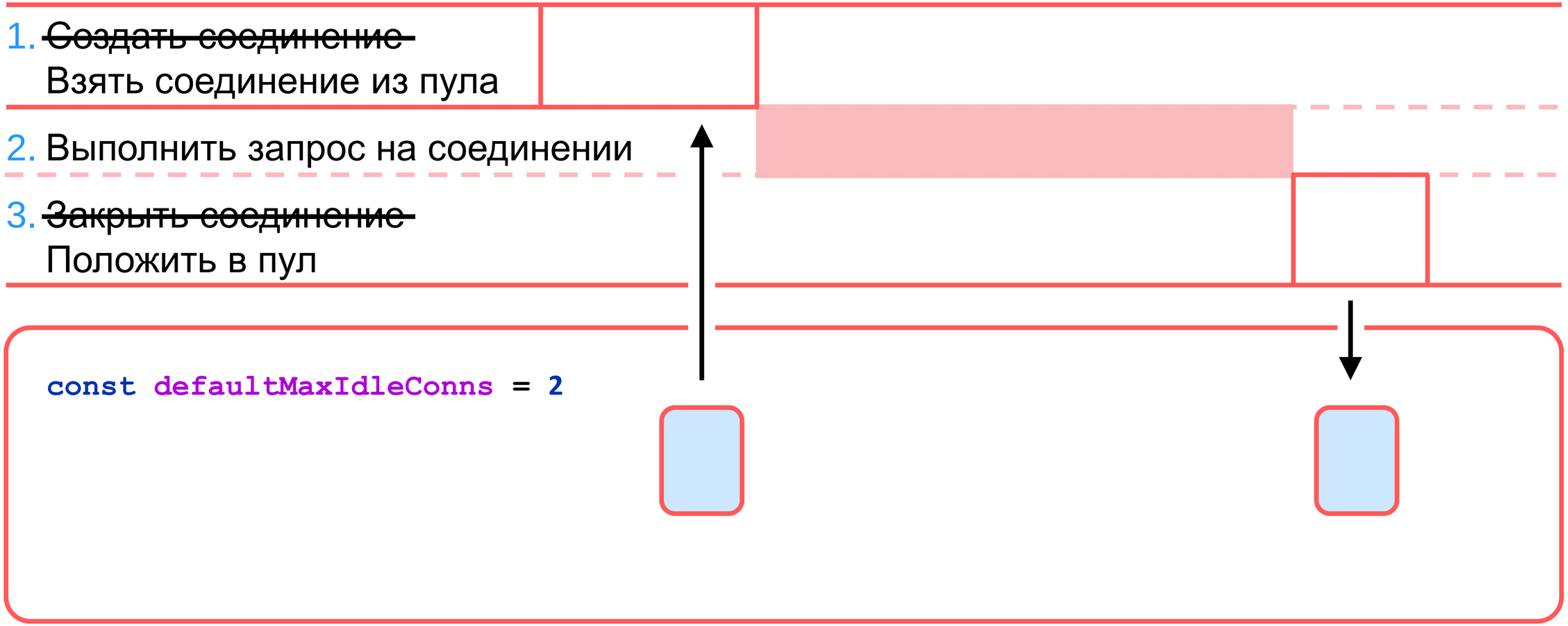
Prepare — обман

github.com/pgjdbc/pgjdbc

clck.ru/36eQA9

Пул соединений в database/sql

1. ~~Создать соединение~~
   Взять соединение из пула
2. Выполнить запрос на соединении
3. ~~Закрыть соединение~~
   Положить в пул



```
const defaultMaxIdleConns = 2
```

Пул соединений в database/sql


```
db.SetConnMaxLifetime(time.Minute)
db.SetMaxIdleConns(10)
db.SetMaxOpenConns(10)
```

exp/sql → database/sql

20.01.2012 7fc4c071
Brad Fitzpatrick
codereview.appspot.com/5536076

driver.ErrBadConn, ретраи


```
func (db *DB) Prepare(query string) (*Stmt, error) {
    var stmt *Stmt
    var err error
    for i := 0; i < 10; i++ {
        stmt, err = db.prepare(query)
        if err != driver.ErrBadConn {
            break
        }
    }
    return stmt, err
}
```

08.03.2012 9fb68a9a
Brad Fitzpatrick
codereview.appspot.com/5785043

Go release 1.0

go.dev/doc/go1

28 марта 2012 года

Первый коммит в YDB

Январь 2014 года

connReuseStrategy


```
const maxBadConnRetries = 2
func (db *DB) retry(fn func(strategy connReuseStrategy) error) error {
  for i := int64(0); i < maxBadConnRetries; i++ {
     err := fn(cachedOrNewConn)
     // retry if err is driver.ErrBadConn
     if err != driver.ErrBadConn {
        return err
     }
  }

  return fn(alwaysNewConn)
}
```

27.03.2015 c468f946
Marko Tiikkaja
clck.ru/36eVHi

Когда встроенный ретраер database/sql не работает

db.retry(...)

```
rows, err := db.Query("SELECT * FROM tbl;")
if err != nil {
  // fallback
}
defer rows.Close()
for rows.Next() {
  // scan row
}
if err := rows.Err(); err != nil {
  // fallback
}
```

no retry

no retry

2015 (go1.5)

Когда встроенный ретраер database/sql не работает

db.retry(...)

```go
tx, err := db.Begin(&sql.TxOptions{})
if err != nil {
  // fallback
}
```

no retry

```go
rows, err := tx.Query("SELECT * FROM tbl;")
if err != nil {
  // fallback
}
defer rows.Close()
```

no retry

```go
for rows.Next() {
  // scan row
}
```

no retry

```go
if err := rows.Err(); err != nil {
  // fallback
}
```

no retry

```go
if err := rows.Commit(); err != nil {
  // fallback
}
```

2015 (go1.5)

context.Context

<a в database/sql ничего не изменилось>

SQL (YQL) в YDB

Именованные параметры в СУБД

- MS SQL
- Oracle
- MySQL
- SQLite3
- IBM DB2
- YDB

Типы параметров в SQL

Литеральные

```
"SELECT * FROM tbl WHERE id=100500 AND year>2012"
```

Позиционные

```
"SELECT * FROM tbl WHERE id=? AND year>?"
```

Нумерованные

```
"SELECT * FROM tbl WHERE id=$1 AND year>$2"
```

Именованные

```
"SELECT * FROM tbl WHERE id=:id AND year>:year"
```

```
"SELECT * FROM tbl WHERE id=$id AND year>$year"
```

sql.Named

```
rows, err := db.Query(
   "SELECT $name, $age",
   sql.Named("$age", 2),
   sql.Named("$name", "Bob"),
)
```

03.10.2016 707a8334
Daniel Theophanes
clck.ru/36gvNn

rows.NextResultSet()

```
for rows.NextResultSet() {
   for rows.Next() {
      // scan row
   }
}
```

06.10.2016 86b2f296
Daniel Theophanes
clck.ru/36gvnM

rows.NextResultSet()


```
rows, err := db.Query(
   "SELECT a1, a2 FROM tblA;"+
   "SELECT b1, b2 FROM tblB;",
)

for rows.NextResultSet() {
   for rows.Next() {
      // scan row
   }
}
```

06.10.2016 86b2f296
Daniel Theophanes
clck.ru/36gvnM

rows.NextResultSet()


```
rows, err := db.Query(
   "SELECT * FROM bigTable;",
)

for rows.NextResultSet() {
   for rows.Next() {
      // scan row
   }
}
```

06.10.2016 86b2f296
Daniel Theophanes
clck.ru/36gvnM

Как работает rows.NextResultSet()?

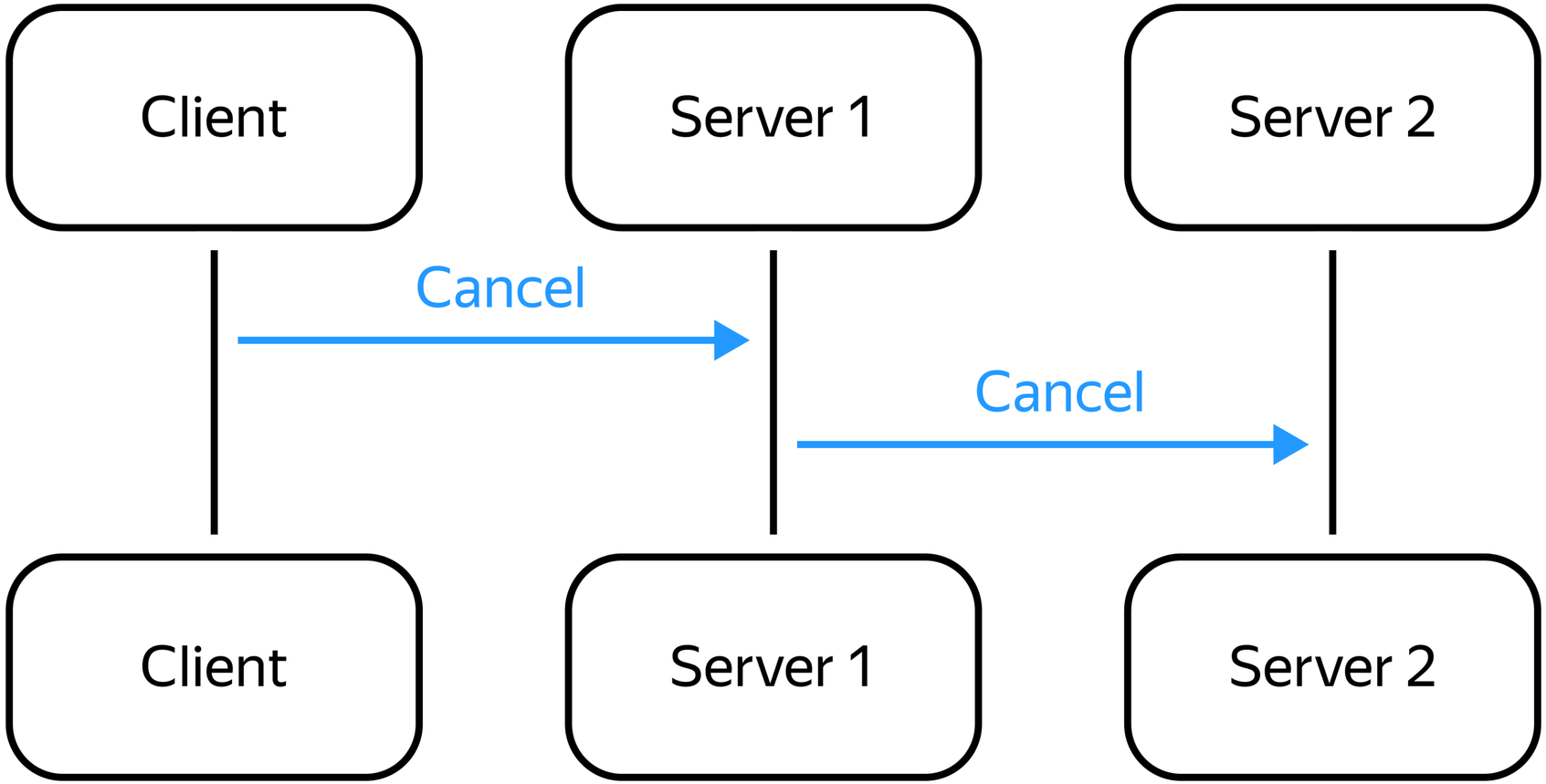

```
type RowsNextResultSet interface {
  Rows

  HasNextResultSet() bool
  NextResultSet() error
}
...
nextResultSet, ok := rs.rowsi.(driver.RowsNextResultSet)
if !ok {
  return false
}
rs.lasterr = nextResultSet.NextResultSet()
if rs.lasterr != nil {
  return false
}
return true
```

sql.*Context

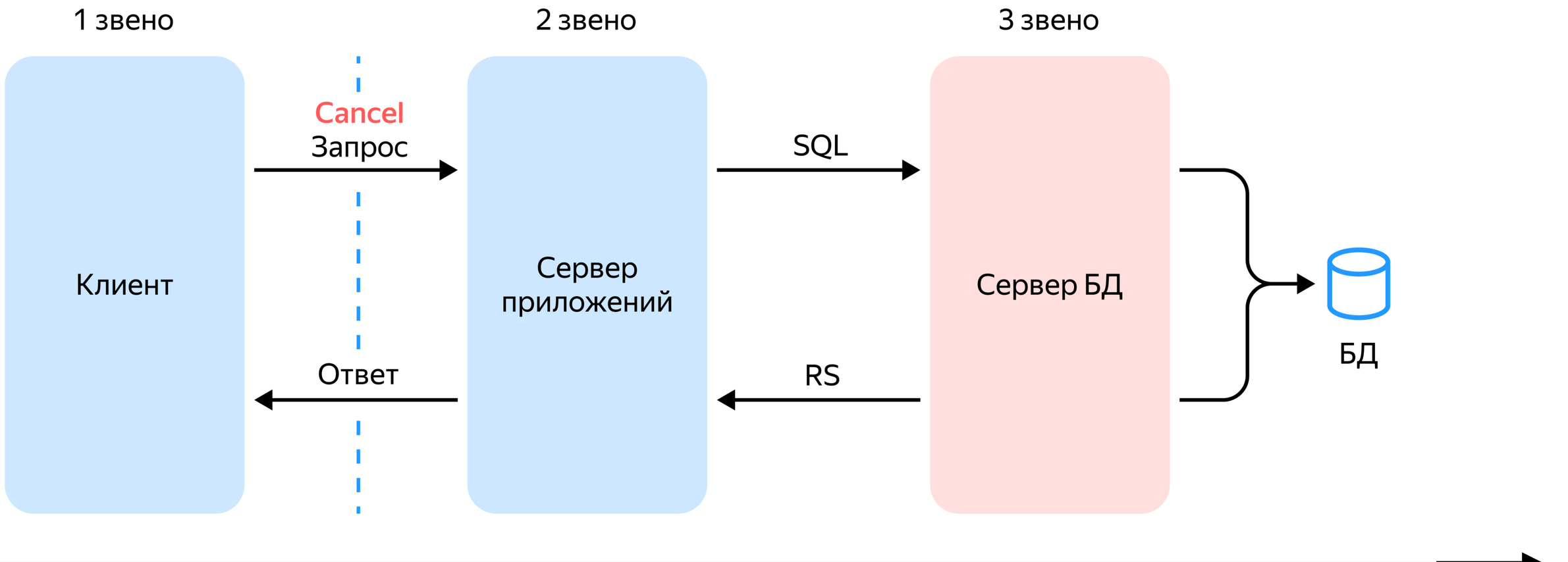
```
db.Exec(query, args...)
db.ExecContext(ctx, query, args...)

db.Query(query, args...)
db.QueryContext(ctx, query, args...)

db.Prepare(query)
db.PrepareContext(ctx, query)

db.Begin()
db.BeginTx(ctx, txOptions)
```

19.09.2016 e13df02e
Daniel Theophanes
clck.ru/36gwF8

Cancel propagation



Если вызывать методы database/sql без контекста



Когда «выпилят» методы без контекста?

Когда «выпилят» методы без контекста?

Russ Cox
14 August 2023
clck.ru/36gwVu

Когда «выпилят» методы без контекста?

Russ Cox
14 August 2023
clck.ru/36gwVu

Никогда

Когда «выпилят» методы без контекста?

Russ Cox
14 August 2023
clck.ru/36gwVu

Никогда,

но можно было бы разметить через Deprecated

«Нативный» go-драйвер для YDB

Сентябрь 2018 года

Sergey Kamardin

GitHub gobwas
Twitter: skamardin
sergey.kamardin.org

Экспериментальный database/ sql driver для YDB

Ноябрь 2018 года

Sergey Kamardin

GitHub gobwas
Twitter: skamardin
sergey.kamardin.org

errors.{Is,As}

До go1.13

```
if err != driver.ErrBadConn {
  // do
}
```

Начиная с go1.13

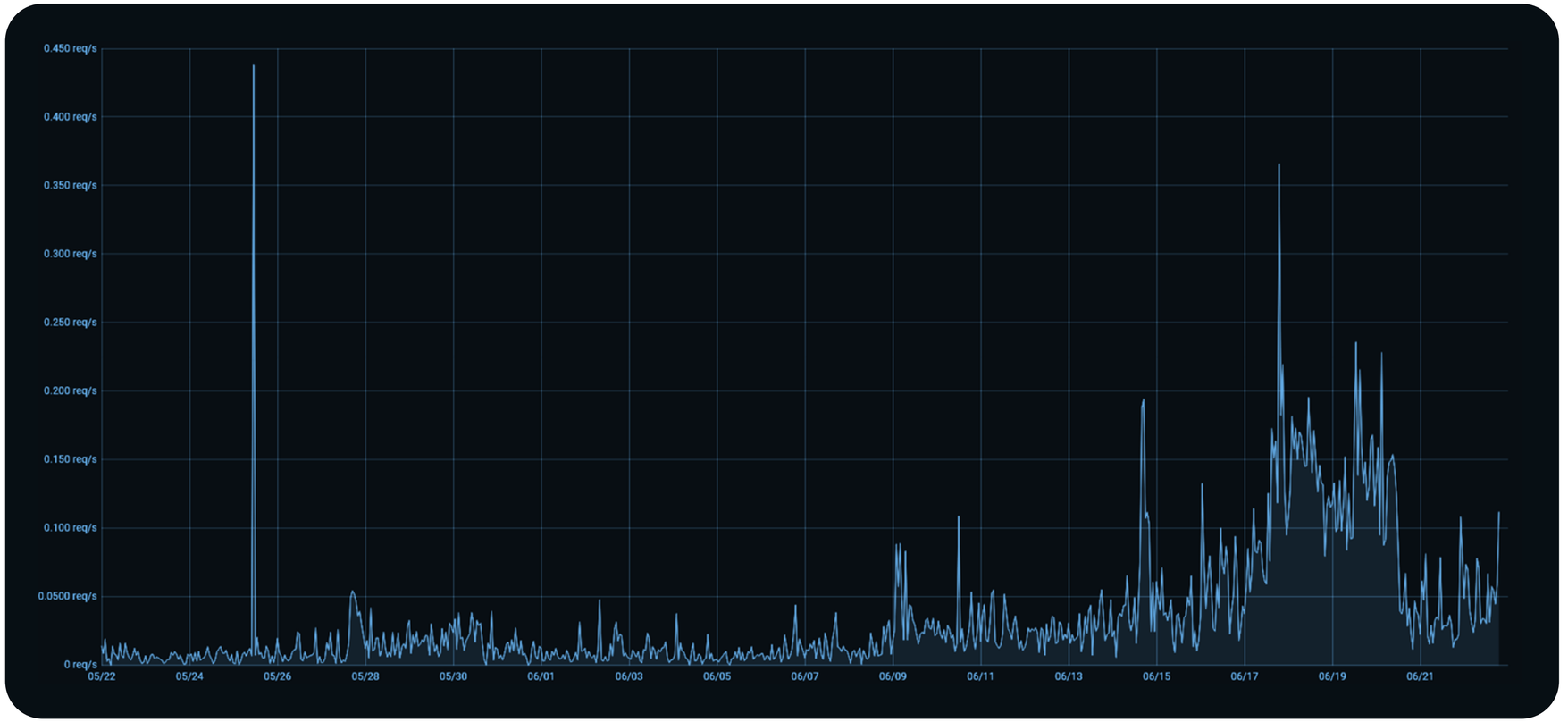
```
if errors.Is(err, driver.ErrBadConn) {
  // do
}
```

2019 (go1.13)

errors.{Is,As}

<a в database/sql ничего не изменилось>

Когда исходная ошибка замалчивается за driver.ErrBadConn



2019 (go1.13)...

...2022 (go1.18)

Транспортные и серверные ошибки

- Canceled
- Unauthenticated
- Internal
- Unknown
- ResourceExhausted
- Unavailable
- InvalidArgument
- FailedPrecondition
- DataLoss
- DeadlineExceeded
- Aborted
- Unimplemented
- AlreadyExists
- OutOfRange
- PermissionDenied

Транспортные ошибки

- Bad_request
- Generic_error
- Unauthorized
- Timeout
- Internal_error
- Bad_session
- Aborted
- Precondition_failed
- Cancelled
- Unavailable
- Already_exists
- Undetermined
- Overloaded
- Not_found
- Unsupported
- Scheme_error
- Session_expired
- Session_busy

Серверные ошибки

clck.ru/36hMw9

С «медленной» экспоненциальной задержкой

Canceled	Unauthenticated	Internal
Unknown	ResourceExhausted	Unavailable
InvalidArgument	FailedPrecondition	DataLoss
DeadlineExceeded	Aborted	Unimplemented
AlreadyExists	OutOfRange	
PermissionDenied		

- Транспортные ошибки
- Серверные ошибки
- Экспоненциальная задержка между попытками от 1 с.

Bad_request	Generic_error	
Unauthorized	Timeout	
Internal_error	Bad_session	
Aborted	Precondition_failed	Cancelled
Unavailable	Already_exists	Undetermined
Overloaded	Not_found	Unsupported
Scheme_error	Session_expired	Session_busy

clck.ru/36hMw9

С «быстрой» экспоненциальной задержкой

Canceled	Unauthenticated	Internal
Unknown	ResourceExhausted	Unavailable
InvalidArgument	FailedPrecondition	DataLoss
DeadlineExceeded	Aborted	Unimplemented
AlreadyExists	OutOfRange	
PermissionDenied		

Транспортные ошибки

Серверные ошибки

Экспоненциальная задержка между попытками от 5 мс.

Bad_request	Generic_error	
Unauthorized	Timeout	
Internal_error	Bad_session	
Aborted	Precondition_failed	Cancelled
Unavailable	Already_exists	Undetermined
Overloaded	Not_found	Unsupported
Scheme_error	Session_expired	Session_busy

clck.ru/36hMw9

Ретраибельные ошибки

- Canceled
- Unknown
- InvalidArgument
- DeadlineExceeded
- AlreadyExists
- PermissionDenied
- Unauthenticated
- ResourceExhausted
- FailedPrecondition
- Aborted
- OutOfRange
- Internal
- Unavailable
- DataLoss
- Unimplemented

- Транспортные ошибки
- Серверные ошибки
- Можно безопасно ретраить
- Можно ретраить, если операция идемпотентная

- Bad_request
- Unauthorized
- Internal_error
- Aborted
- Unavailable
- Overloaded
- Scheme_error
- Generic_error
- Timeout
- Bad_session
- Precondition_failed
- Already_exists
- Not_found
- Session_expired
- Cancelled
- Undetermined
- Unsupported
- Session_busy

clck.ru/36hMw9

Следует удалить сессию

Canceled	Unauthenticated	Internal
Unknown	ResourceExhausted	Unavailable
InvalidArgument	FailedPrecondition	DataLoss
DeadlineExceeded	Aborted	Unimplemented
AlreadyExists	OutOfRange	
PermissionDenied		

Транспортные ошибки

Серверные ошибки

Сессия более непригодна

Bad_request	Generic_error	
Unauthorized	Timeout	
Internal_error	Bad_session	
Aborted	Precondition_failed	Cancelled
Unavailable	Already_exists	Undetermined
Overloaded	Not_found	Unsupported
Scheme_error	Session_expired	Session_busy

clck.ru/36hMw9

Пессимизация соединений YDB

- Canceled
- Unknown
- InvalidArgument
- DeadlineExceeded
- AlreadyExists
- PermissionDenied
- Unauthenticated
- ResourceExhausted
- FailedPrecondition
- Aborted
- OutOfRange
- Internal
- Unavailable
- DataLoss
- Unimplemented

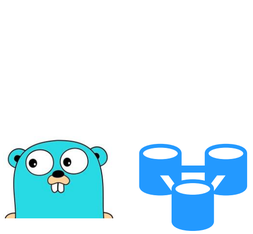
Транспортные ошибки

Серверные ошибки

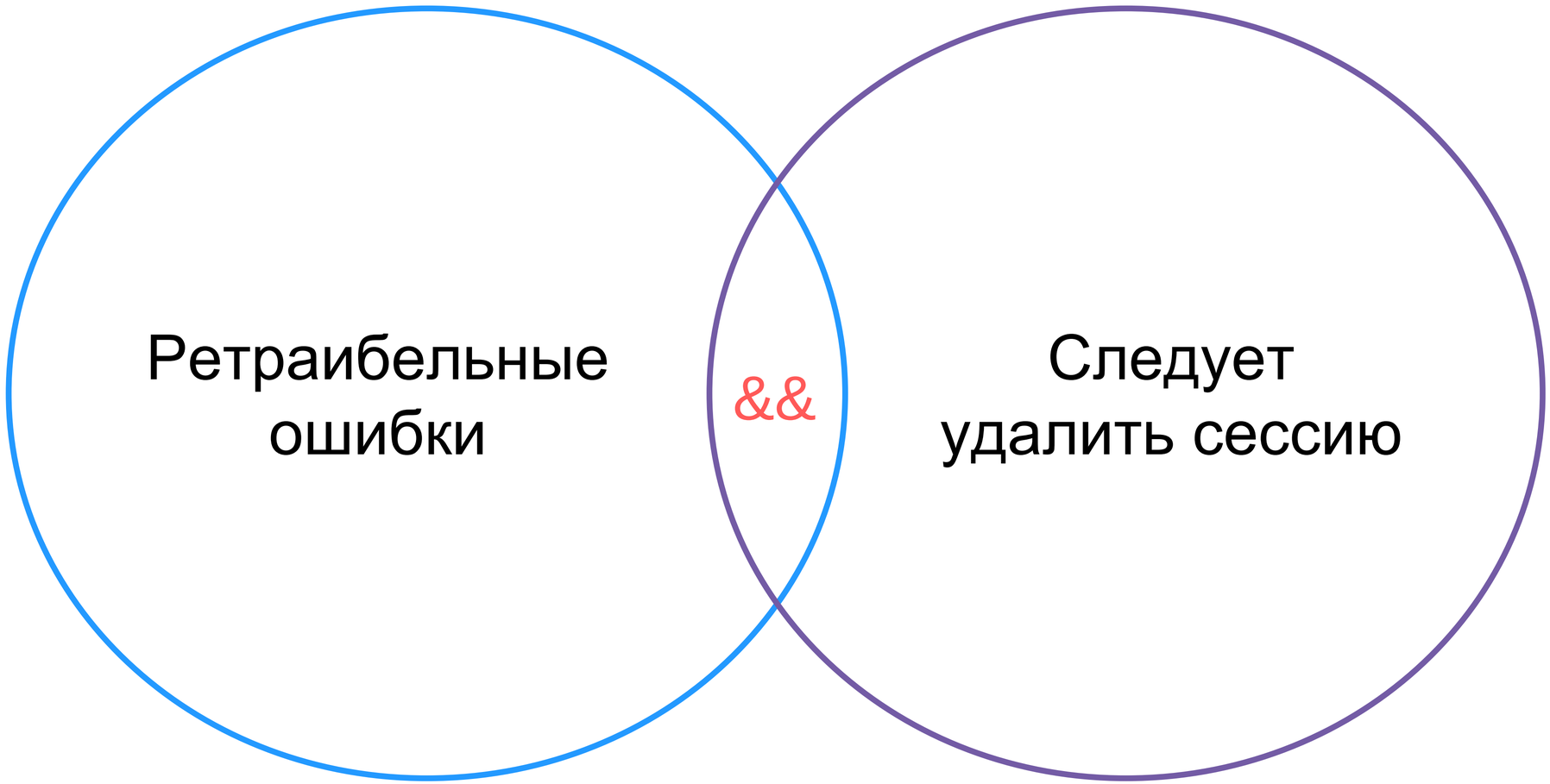
Следует «забыть» про все сессии к ноде YDB

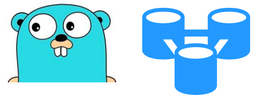
- Bad_request
- Unauthorized
- Internal_error
- Aborted
- Unavailable
- Overloaded
- Scheme_error
- Generic_error
- Timeout
- Bad_session
- Precondition_failed
- Already_exists
- Not_found
- Session_expired
- Cancelled
- Undetermined
- Unsupported
- Session_busy

clck.ru/36hMw9

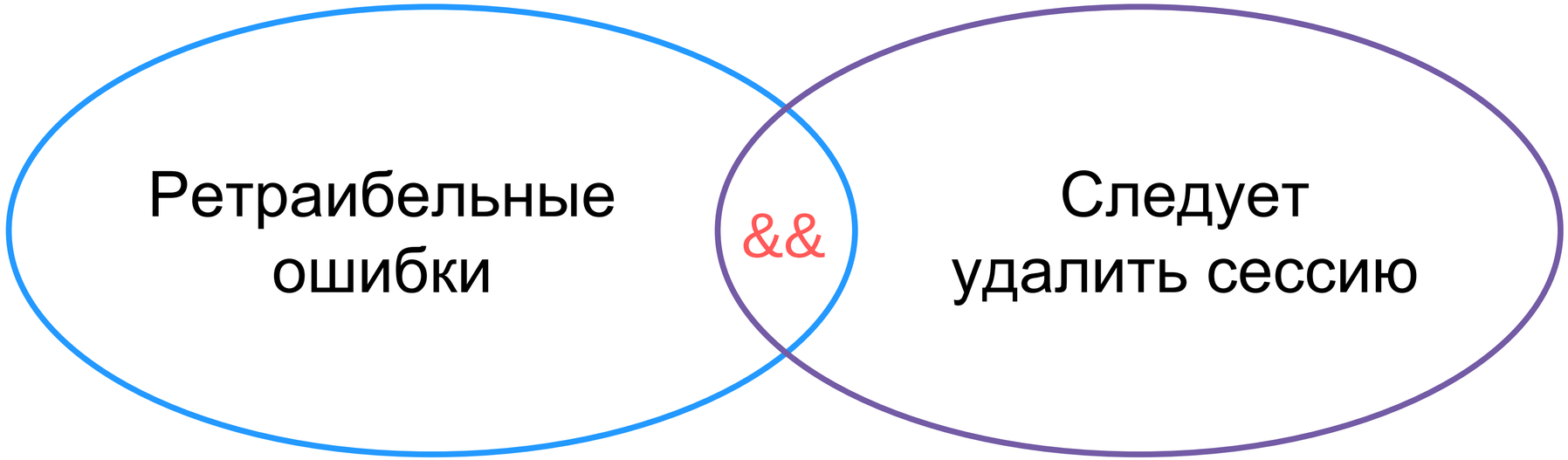


driver.ErrBadConn



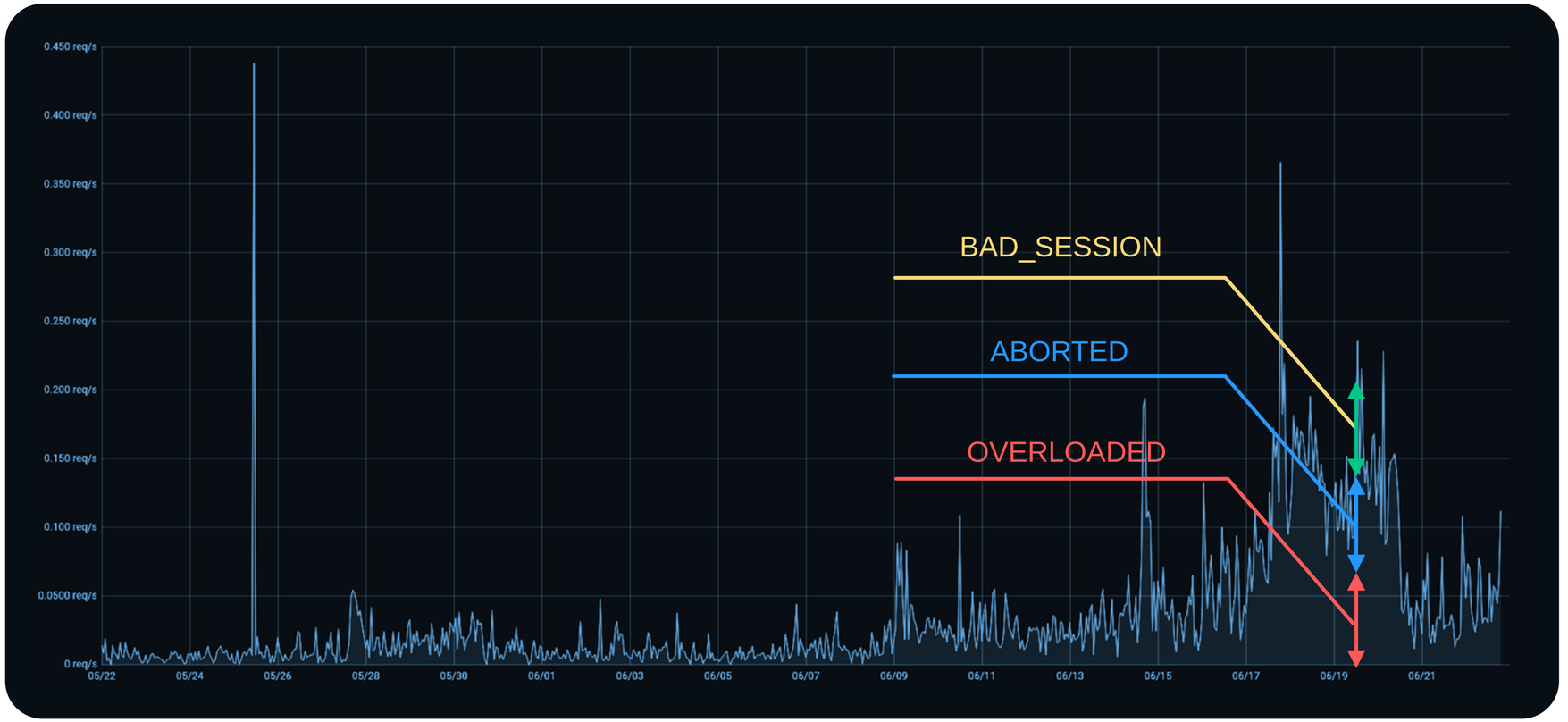


driver.ErrBadConn



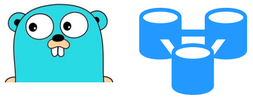

```
//go:build !go1.18
// +build !go1.18
...
if mustDeleteSession(sourceErr) && mustRetry(sourceErr) {
  return driver.ErrBadConn
}
```

Если бы driver.ErrBadConn не «замалчивал» исходную ошибку



2019 (go1.13)...

...2022 (go1.18)

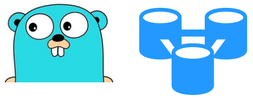


Нативный драйвер YDB

v3.0.0
23.10.2021 1c67c419
Aleksey Myasnikov
clck.ru/36gxff


«Умные» ретраеры в нативном драйвере YDB


```
err := db.Table().Do(ctx, func(ctx context.Context, s table.Session) error {
   tx, result, err := s.Execute(ctx, txControl, query,
      table.NewQueryParameters(params...))
   if err != nil {
      return err
   }
   var title, content string
   for result.NextResultSet(ctx) {
      for result.NextRow() {
         if err := result.Scan(&title, &content); err != nil {
            return err
         }
         log.Println(title, content)
      }
   }
   return result.Err()
}, table.WithIdempotent())
```



Официальный database/sql-драйвер для YDB

v3.33.0
10.08.2022 b0a755c
Aleksey Myasnikov
clck.ru/36gy7Y

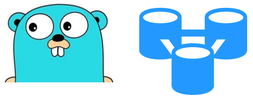
«Умные» ретраеры YDB для database/sql


```
err = retry.Do(ctx, db,
    func(ctx context.Context, cc *sql.Conn) error {
        ...
    },
    retry.WithIdempotent(true),
)

err := retry.DoTx(ctx, db,
    func(ctx context.Context, tx *sql.Tx) error {
        ...
    },
    retry.WithIdempotent(true),
    retry.WithTxOptions(&sql.TxOptions{
        Isolation: sql.LevelSnapshot,
        ReadOnly: true,
    }),
)
```

errors.Is(err, driver.ErrBadConn)


```go
const maxBadConnRetries = 2
func (db *DB) retry(fn func(strategy connReuseStrategy) error) error {
 for i := int64(0); i < maxBadConnRetries; i++ {
    err := fn(cachedOrNewConn)
    // retry if err is driver.ErrBadConn
    if err == nil || !errors.Is(err, driver.ErrBadConn) {
       return err
    }
 }
 return fn(alwaysNewConn)
}
```

12.07.2021 1f368d5b
Daniel Theophanes
clck.ru/36gy7Y



Когда из driver.ErrBadConn невозможно получить исходную ошибку

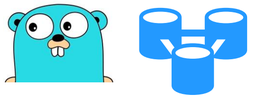
Было


```go
//go:build !go1.18
// +build !go1.18
...
if isBadConnErr(sourceErr) {
  return driver.ErrBadConn
}
```

2021 (go1.17)

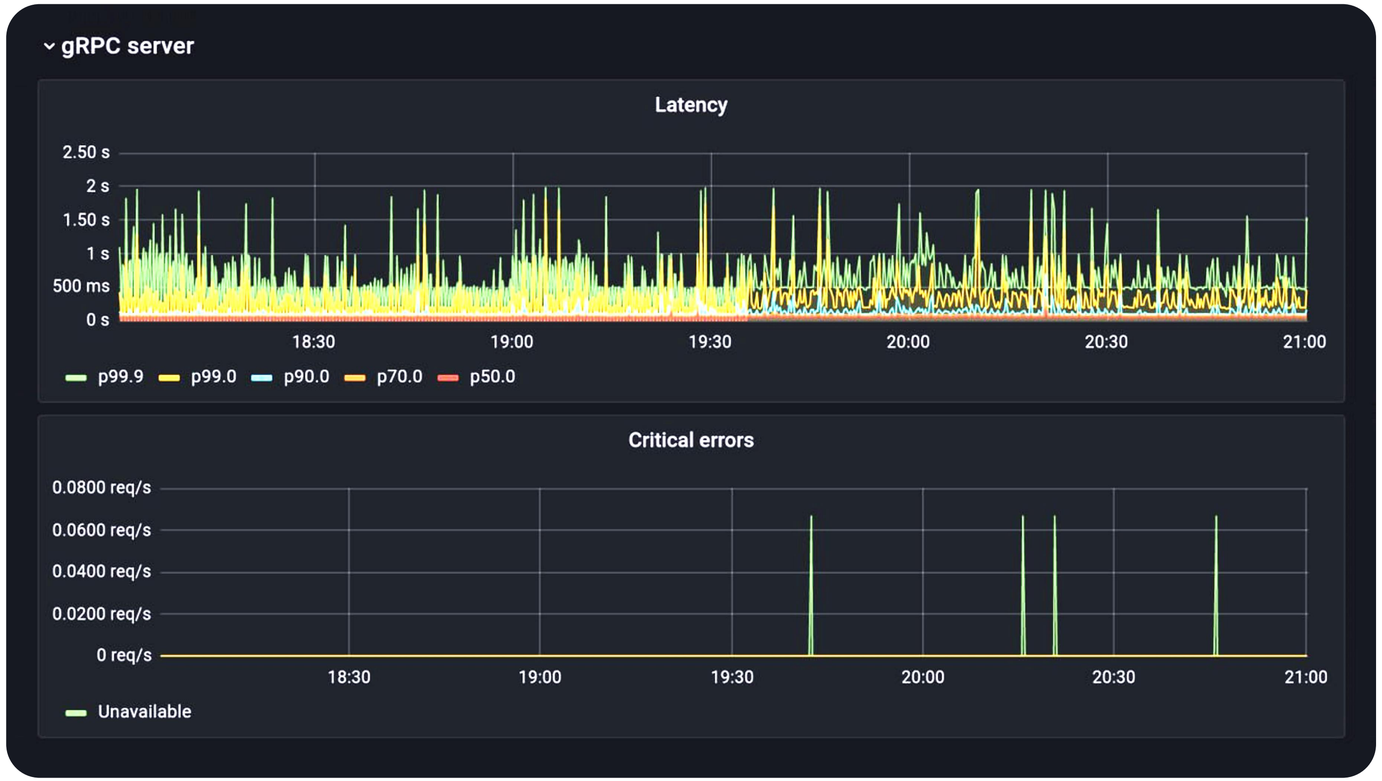

driver.ErrBadConn


```go
//go:build go1.18
// +build go1.18

type badConnErr struct {
    sourceErr error
}

func (e badConnErr) Is(err error) bool {
    if err == driver.ErrBadConn {
      return true
    }
    return errors.Is(e.sourceErr, err)
}
```

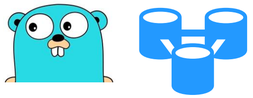


Когда из driver.ErrBadConn можно получить исходную ошибку

Стало




```go
//go:build go1.18
// +build go1.18
...
if isBadConnErr(err) {
  return badConnErr{
    sourceErr: err,
  }
}
```

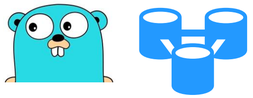


Биндинги запросов


```
SELECT id, name, salary
FROM staff
WHERE
 department_id = ? AND
 salary > ?;
```

```
-- bind TablePathPrefix
PRAGMA TablePathPrefix("/local/path/to/my/folder");

-- bind declares
DECLARE $p0 AS Int32;
DECLARE $p1 AS Double;

-- origin query with positional args replacement
SELECT id, name, salary
FROM staff
WHERE
 department_id = $p0 AND
 salary > $p1;
```

v3.44.0
03.04.2023 f7f81714
Aleksey Myasnikov
clck.ru/36h2uc



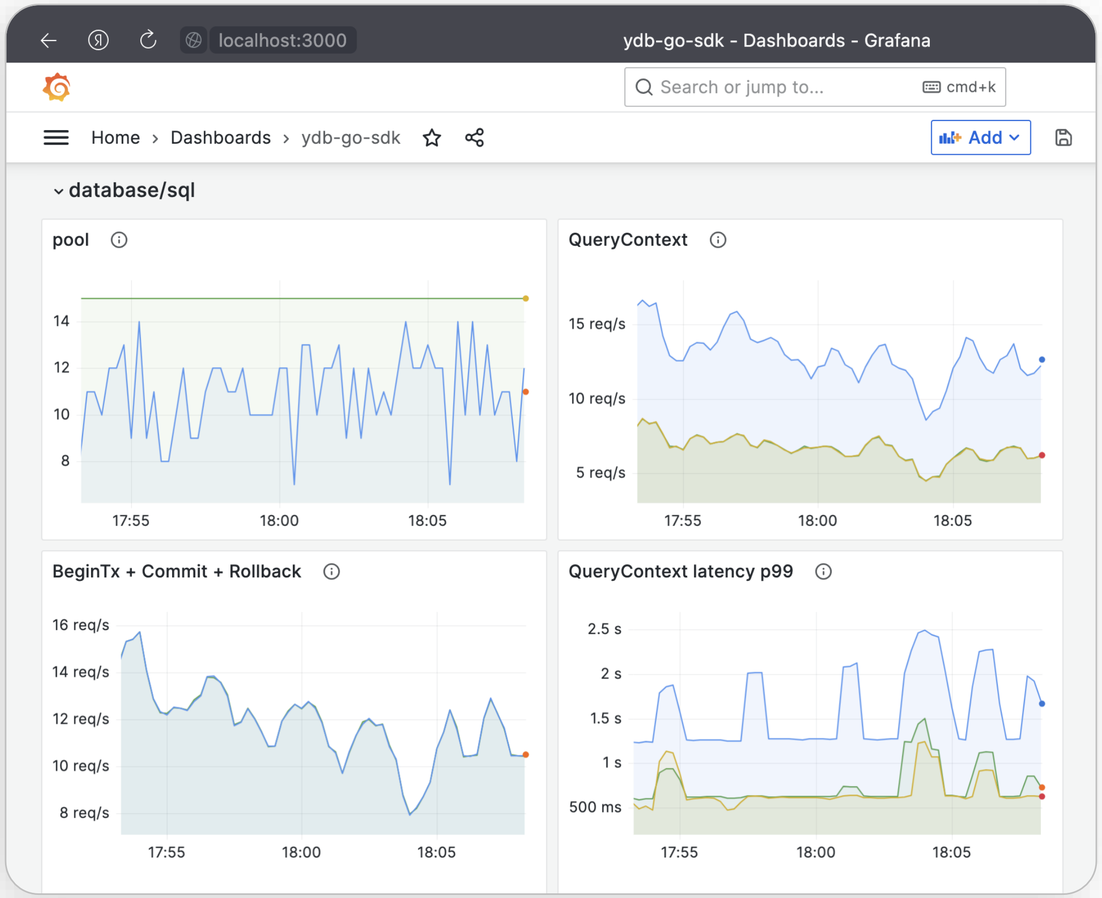
Поддержка логирования событий database/sql-драйвера

github.com/ydb-platform/ydb-go-sdk-zap

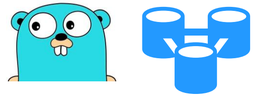
github.com/ydb-platform/ydb-go-sdk-zerolog

github.com/ydb-platform/ydb-go-sdk-logrus


Метрики database/sql-драйвера



github.com/ydb-go-sdk-prometheus



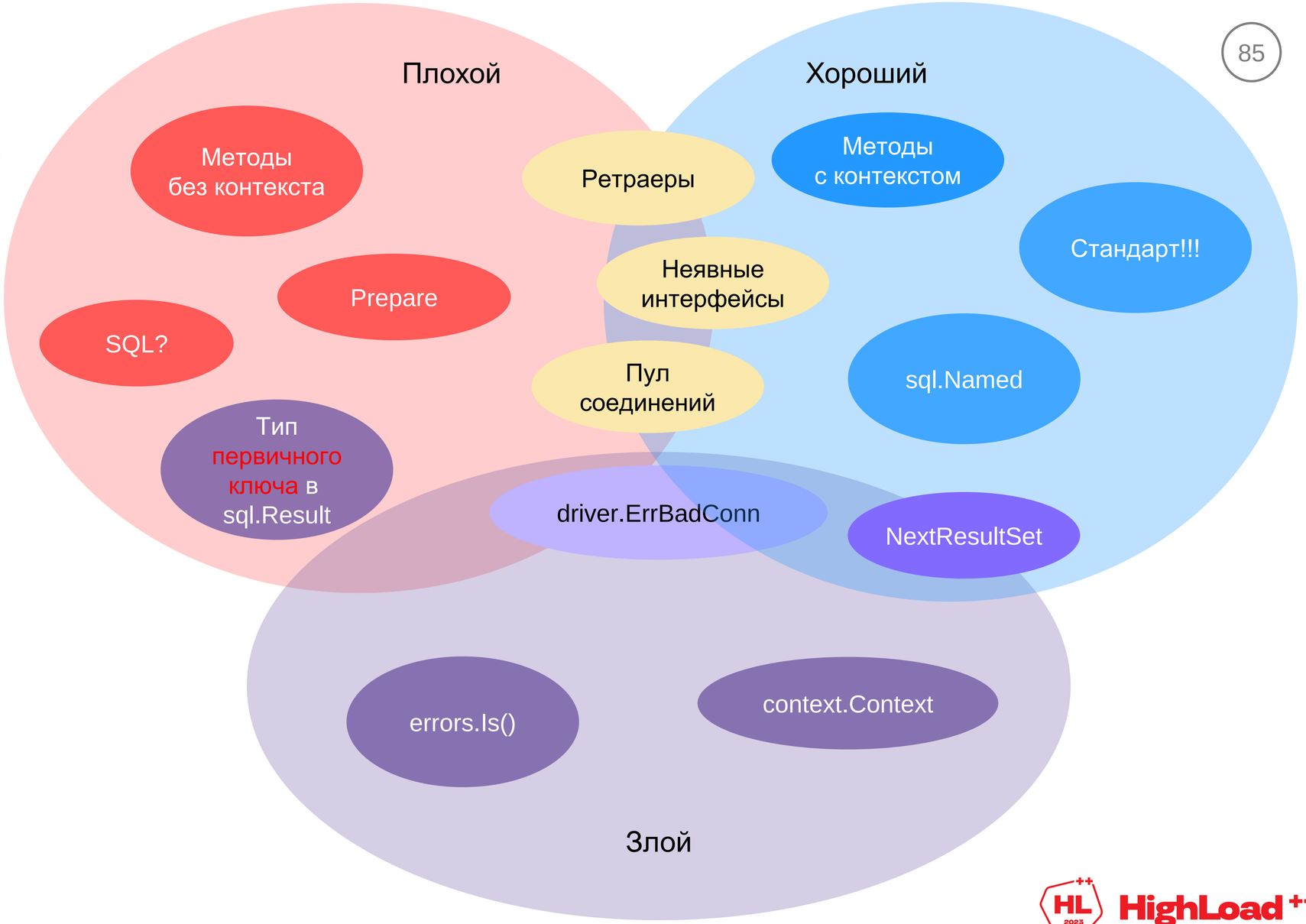
Трассировка database/sql-драйвера


```
main main
  main github.com/ydb-platform/ydb-go-sdk/v3.Open
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.NewPool
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/balancer.New
      main github.com/ydb-platform/ydb-go-sdk/v3/retry.Retry
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/balancer.(*Balancer).clusterDiscoveryAttempt
          main github.com/ydb-platform/ydb-go-sdk/v3/internal/discovery.(*Client).Discover
            main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
              main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).realConn
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/table.newClient
  main prepareSchema(series)
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/xsql.(*Connector).Connect
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/table.(*Client).CreateSession
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/table.newSession
          main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
            main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).realConn
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/table.(*session).Close
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/xsql.(*conn).IsTableExists
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/scheme.(*Client).ListDirectory
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/scheme.(*Client).ListDirectory
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/scheme.(*Client).ListDirectory
        main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/xsql.(*conn).execContext
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/xsql.(*conn).execContext
      main github.com/ydb-platform/ydb-go-sdk/v3/internal/conn.(*conn).Invoke
  main prepareSchema(seasons)
    main github.com/ydb-platform/ydb-go-sdk/v3/internal/xsql.(*conn).IsTableExists
```

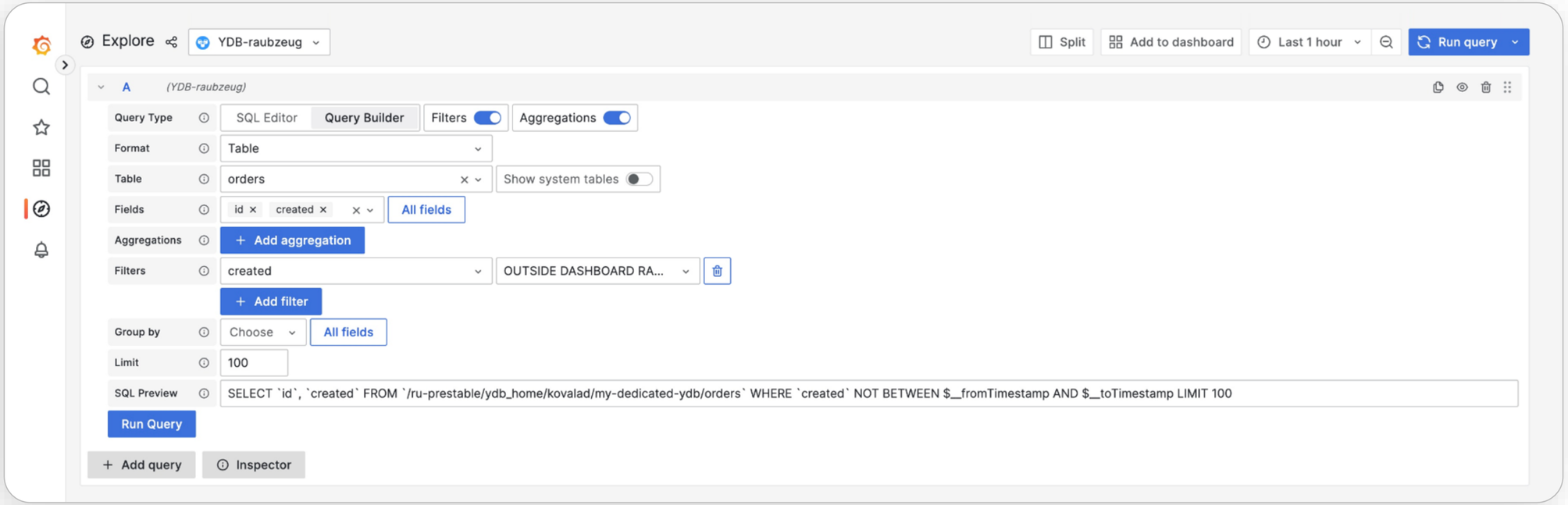
github.com/ydb-go-sdk-opentracing

github.com/ydb-go-sdk-otel

Плохой, хороший, злой…



ydb-grafana-datasource-plugin



21.09.2023
Elena Makarova
github.com/ydb-platform/ydb-grafana-datasource-plugin


Поддержка YDB в goose (v3.16.0+)


```
foo@bar:~$ goose -dir migrations -table
goose_db_version ydb
"grpc://localhost:2136/local?go_query_mode=scripting&
go_fake_tx=scripting&go_query_bind=declare,numeric" up
```

25.10.2023 0c243fb
Aleksey Myasnikov
github.com/pressly/goose/pull/592


Другие активности поверх database/sql-драйвера для YDB (WIP)

github.com/ydb-platform/go-ycsb (@asmyasnikov)

github.com/ydb-platform/squirrel (@Loganche)

label:student-projects
clck.ru/36h8ct

github.com/ydb-platform/gorm-driver (@asmyasnikov, @ImpressionableRaccoon, @rekby)

github.com/ydb-platform/xorm (@datbeohbbh)

Голосуйте за мой доклад

Алексей Мясников,
Руководитель AppTeam YDB,
Яндекс, YDB

t.me/asmyasnikov
github.com/asmyasnikov
habr.com/ru/users/asmyasnikov
medium.com/@asmyasnikov

Скачать этот доклад: clck.ru/36h9SF

Golang Conf 2023