



Новые возможности СУБД Ред База Данных 3.0

Роман Симаков

директор департамента развития системных продуктов

Предисловие

- ✓ 2016 - Firebird 3.0
- ✓ 2017 - Ред База Данных 3.0
- ✓ Портировались доработки от Ред База Данных 2.5 в 3.0 и попутно в Firebird 3.0 и master ветку.
- ✓ Влиты основные доработки:
 - ✓ Контроль доступа к DDL операциям
 - ✓ Возможность назначения роль другой роли
 - ✓ Запуск процедур и функций в контексте влательца

Предисловие

- ✓ Для Ред База Данных 3.0 это не новое и потому не говорим
- ✓ Перешли на gitlab и ветки образуются параллельно с Firebird, что позволяет больше не отставать с релизами.
- ✓ Многие перерабатывались с учетом новой архитектуры:
 - ✓ Плагины аутентификации
 - ✓ политики доступа пользователей

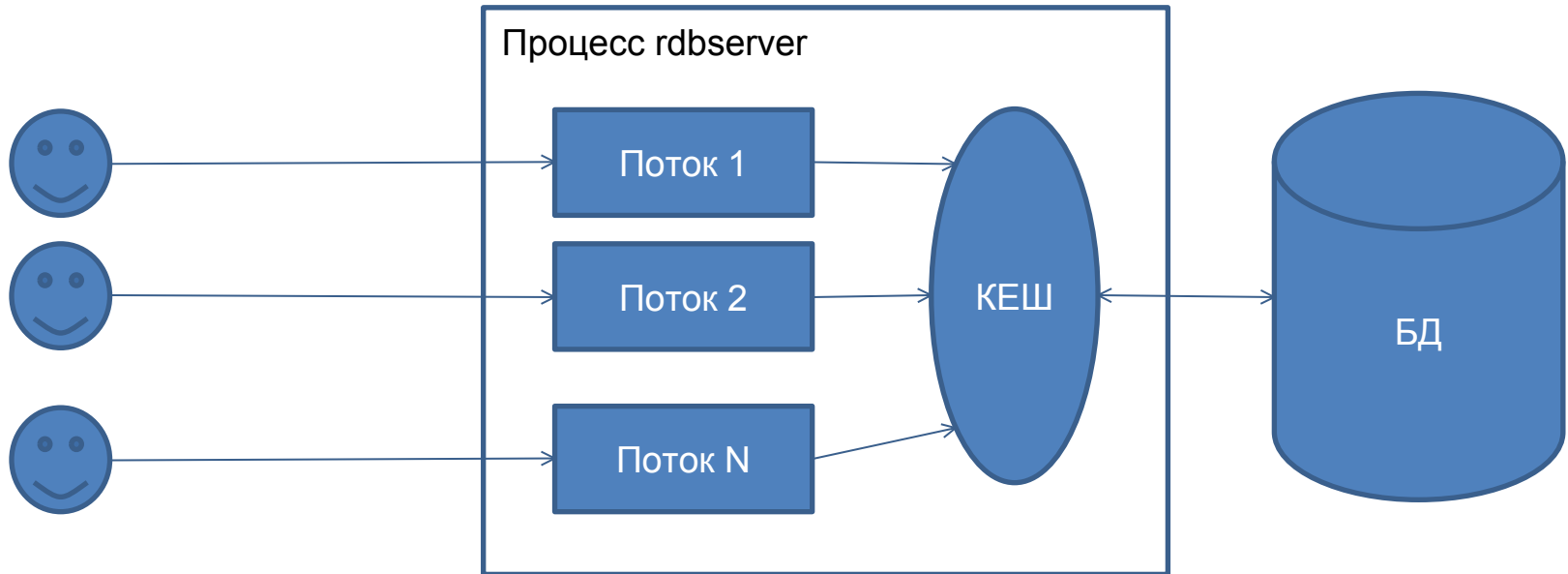
Обзор

- ✓ Изменения в архитектуре сервера
- ✓ Новые возможности языка SQL
- ✓ Новые методы аутентификации
- ✓ Прочие изменения

Унифицированы исполняемые файлы

- ✓ Ядро СУБД - универсальная библиотека, поддерживающая одну версию ODS (On Disk Structure)
- ✓ Может линковаться в процесс пользователя как embedded версия
- ✓ Может использоваться "сетевым слушателем"
- ✓ Один серверный процесс для всех архитектур (rdbserver) вместо разных (rdb_inet_server, rdb_smp_server, rdbserver)
- ✓ В файле конфигурации можно выбрать режим работы (Super, SuperClassic, Classic)
- ✓ Проще собирать пакеты

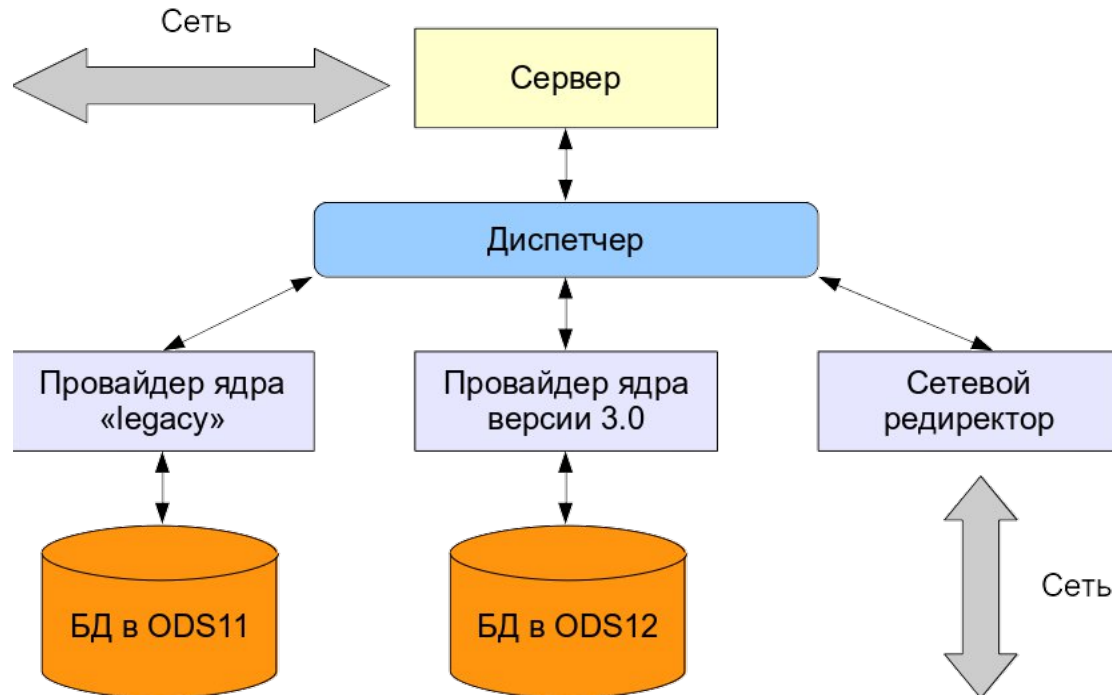
Многопоточная архитектура SUPERSERVER с общим кешем



Плагины

- ✓ Доступны следующие виды плагинов:
 - ✓ Плагины аутентификации
 - ✓ Внешние движки
 - ✓ Плагины трассировки
 - ✓ Плагины шифрования
 - ✓ Провайдеры

Провайдеры



Внешние движки

- ✓ Через стандартное API реализуют User Defined Routines
- ✓ Позволяют реализовать:
 - ✓ Процедуры
 - ✓ Функции
 - ✓ Триггеры
- ✓ Реализовывать можно на любом языке программирования
- ✓ Java из коробки

Примеры объявления UDR

```
create procedure gen_rows (start_n integer not null, end_n integer not null)  
  returns (n integer not null)  
  external name 'udrcpp_example!gen_rows' engine udr;
```

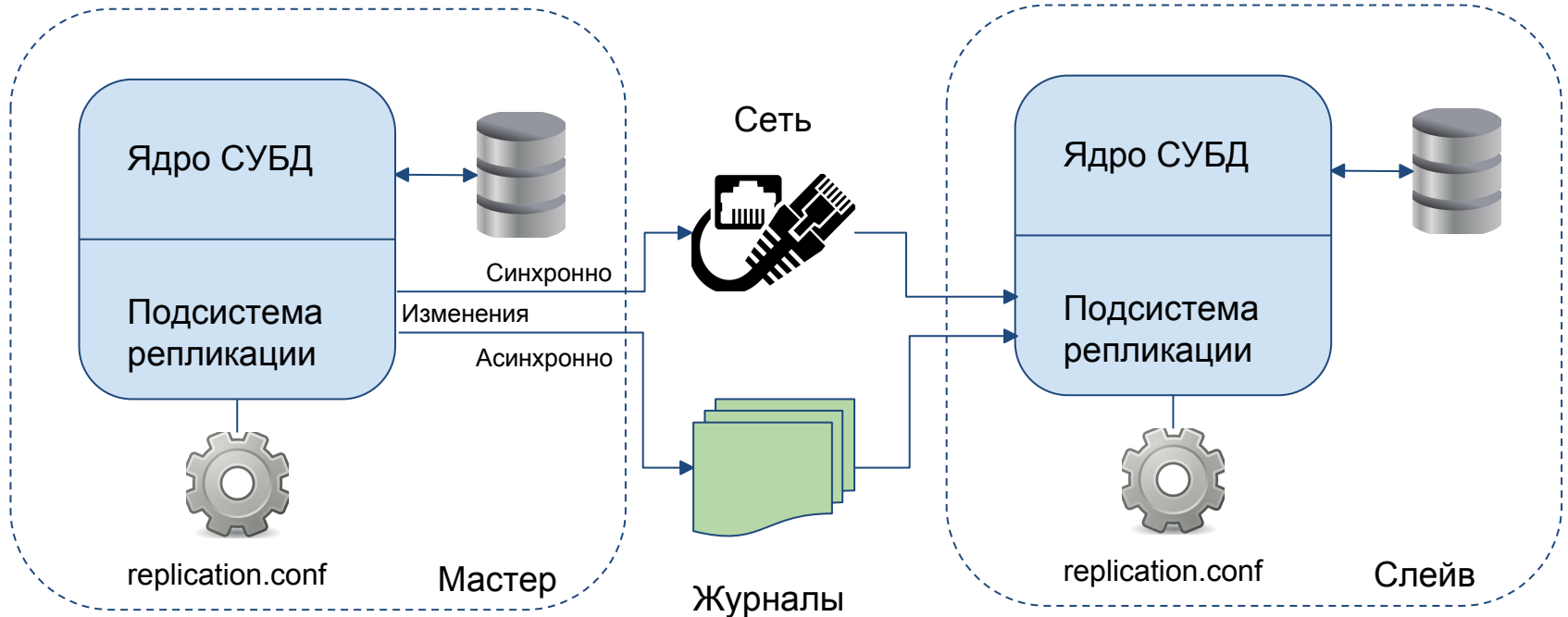
```
create function wait_event (event_name varchar(31) character set ascii)  
  returns integer  
  external name 'udrcpp_example!wait_event'  
  engine udr;
```

```
create trigger persons_replicate  
  after insert on persons  
  external name 'udrcpp_example!replicate!ds1'  
  engine udr;
```

Встроенная репликация

- ✓ Режимы: синхронный, полусинхронный*, асинхронный
- ✓ Возможность push- и pull-подходов к передаче архивных сегментов
- ✓ Многоцелевая и каскадная репликация
- ✓ Доступность реплики на чтение
- ✓ Возможен мульти-мастер*

Схема встроенной репликации



DDL триггеры

```
CREATE EXCEPTION EX_BAD_SP_NAME  
'Name of procedures must start with '@1' : '@2''';  
  
CREATE TRIGGER TRG_SP_CREATE BEFORE CREATE PROCEDURE  
AS  
DECLARE SP_NAME VARCHAR(255);  
BEGIN  
    SP_NAME = RDB$GET_CONTEXT('DDL_TRIGGER', 'OBJECT_NAME');  
  
    IF (SP_NAME NOT STARTING 'SP_')  
        THEN EXCEPTION EX_BAD_SP_NAME USING ('SP_', SP_NAME);  
END;
```

SQL - функции

```
CREATE FUNCTION F (X INT) RETURNS INT  
AS  
BEGIN  
    RETURN X+1;  
END;  
  
SELECT F (5) FROM RDB$DATABASE;
```

SQL - подпрограммы

```
EXECUTE BLOCK RETURNS (N INT)
AS
  DECLARE FUNCTION F(X INT) RETURNS INT
  AS
  BEGIN
    RETURN X+1;
  END;
BEGIN
  N = F(5);
  SUSPEND;
END
```

Пакеты

```
CREATE OR ALTER PACKAGE TEST -- package header, declarations only
AS
BEGIN
  PROCEDURE P1(I INT) RETURNS (O INT); -- public procedure
END
```

```
RECREATE PACKAGE BODY TEST -- package body, implementation
AS
BEGIN
  FUNCTION F1(I INT) RETURNS INT; -- private function
  PROCEDURE P1(I INT) RETURNS (O INT)
  AS
  BEGIN
  END;
  FUNCTION F1(I INT) RETURNS INT
  AS
  BEGIN
  RETURN 0;
  END;
END
```


Оконные функции

```

SELECT A, B, C,
       SUM(C) OVER(),
       SUM(C) OVER(ORDER BY A, B),
       SUM(C) OVER(PARTITION BY A),
       SUM(C) OVER(PARTITION BY A ORDER BY B)

```

A	B	C	SUM	SUM1	SUM2	SUM3
1	1	30	141	30	60	30
1	2	20	141	50	60	50
1	3	10	141	60	60	60
2	1	25	141	85	40	25
2	2	15	141	100	40	40
3	1	41	141	141	41	41

Полная поддержка MERGE по стандарту SQL-2008

```
MERGE INTO <table>  
  USING <table_or_join>  
    ON <search_condition>  
  [WHEN MATCHED [AND <search_condition>] THEN  
    UPDATE SET col1 = val1, ..., colN = valN  
  |  
    DELETE]  
  [WHEN NOT MATCHED [AND <search_condition>] THEN  
    INSERT [(col1, ..., colN)] VALUES (val1, ..., valN)]
```

Множественные части **WHEN**, использование **DELETE** и **RETURNING**

MERGE INTO TABLE

USING LOG

ON TABLE.PK = LOG.PK

**WHEN MATCHED AND LOG.ACTION = 'D' THEN
DELETE]**

WHEN MATCHED THEN

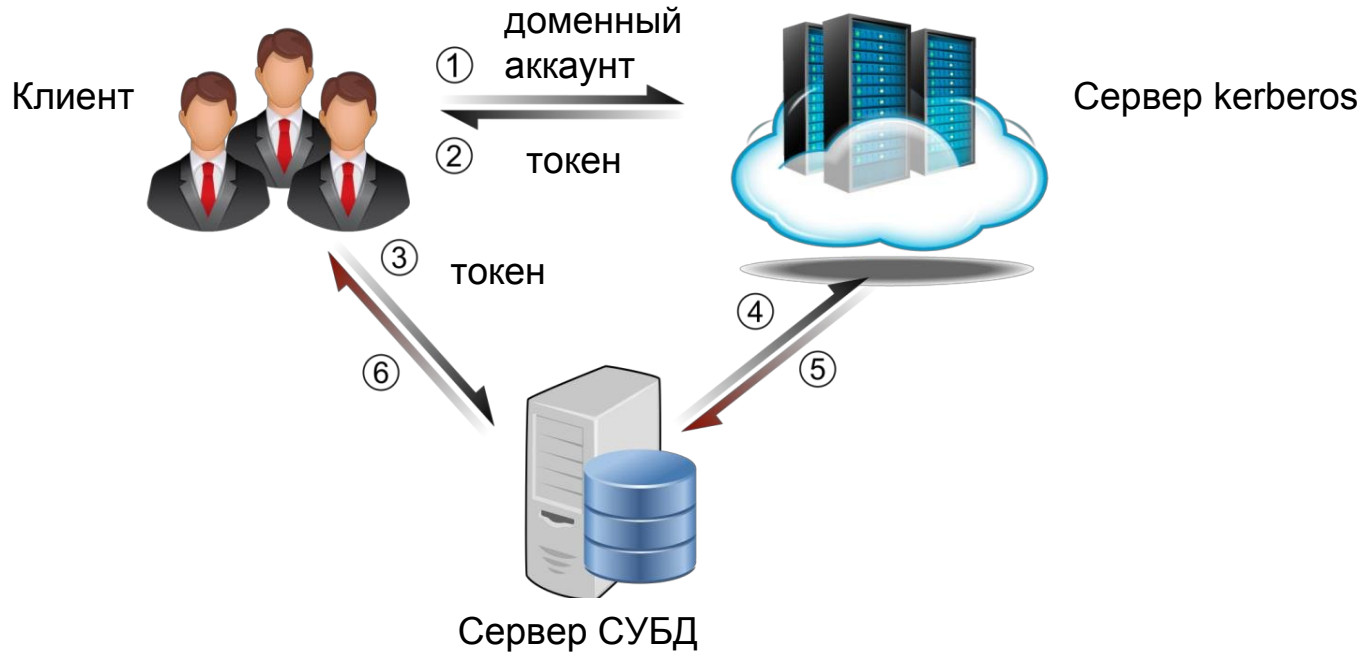
UPDATE SET col1 = LOG.val1, ...

WHEN NOT MATCHED THEN

INSERT (col1, ...) VALUES (val1, ...)

RETURNING colN,... INTO :varN, ...

Аутентификация через GSS API





Защищенная редакция СУБД Ред База Данных 3.0 завершает сертификацию на соответствие требованиям (получено положительное заключение лаборатории):

5 класса защищенности по РД «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации»

4 уровню контроля по РД «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей».

Может применяться в государственных информационных системах (ГИС) до **первого класса** защищенности включительно и для защиты информации в информационных системах персональных данных (ИСПДн) до **первого уровня** защищенности персональных данных включительно.

Прочие изменения

- ✓ Двухнаправленные курсоры
- ✓ Тип данных BOOLEAN и операции с ним
- ✓ Уникальные столбцы
- ✓ Database linger
- ✓ Максимальный размер файла 64Тб вместо 32Тб
- ✓ Количество транзакций 2^{48} вместо 2^{32} (расходы после 2^{32})
- ✓ SCN страницы для инкрементального бэкапа

Спасибо за внимание!



См. также: www.reddatabase.ru

Facebook: www.fb.com/reddatabase

Telegram: <https://t.me/reddatabase>