Лекции 1-2

1. Архитектура СУБД

ИТМО ВТ

Архитектура ANSI-SPARC

Предложена в 1975 г. подкомитетом SPARC (Standards Planning And Requirements Commitee) ANSI.

Архитектура СУБД включает в себя 3 уровня:

- Внешний (пользовательский).
- Промежуточный (концептуальный).
- Внутренний (физический).

Почти все современные СУБД соответствуют принципам ANSI-SPARC.

ИТМО ВТ

СУБД PostgreSQL

- Свободно-распространяемая СУБД на основе языка SQL.
- По классификации — *объектно-реляционная клиент-серверная СУБД*.
- Разработка (Postgres) началась в 1986 году.
- Postgres95 — введен язык SQL.
- С 1996 — PostgreSQL.
- Актуальная версия — 14.1.

ИТМО ВТ

Структуры СУБД

Структуру СУБД можно рассматривать по-разному:

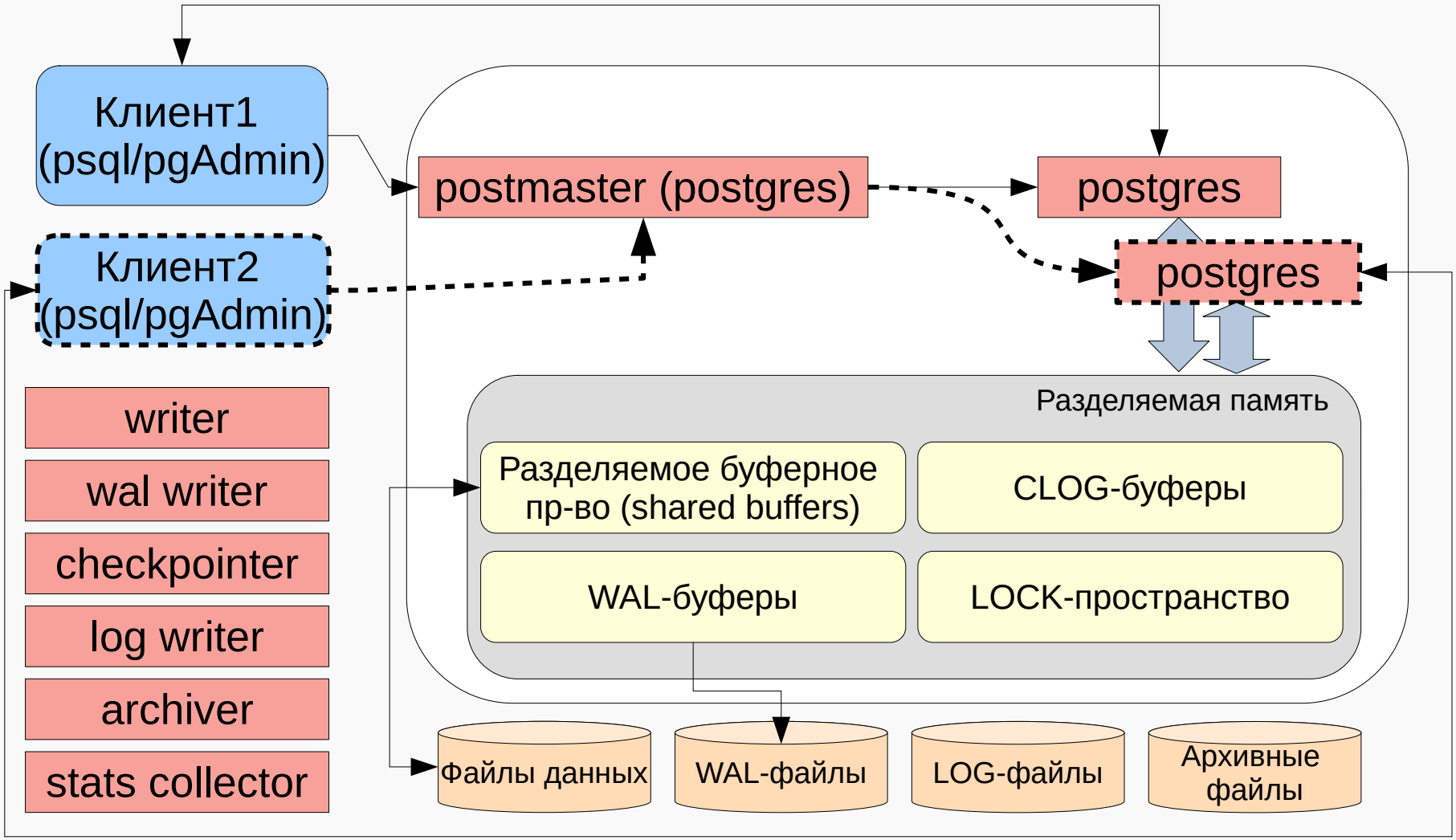
- На уровне структур в основной памяти ЭВМ.
- На уровне процессов в ОС.
- На уровне структуры хранилища данных в ФС.

Кластер БД — множество баз данных, управляемых экземпляром сервера БД (PostgreSQL).

Инстанс (экземпляр) — процессы СУБД + разделяемая память.

ИТМО ВТ

Архитектура PostgreSQL



ИТМО ВТ

Разделяемое буферное пространство (Shared buffers)

- Хранит копии блоков данных (страниц), считанных из файлов данных.
- Служит для минимизации числа операций обмена с диском.
- Если нужного блока (страницы) данных нет в SB, он читается с диска и помещается в SB.
- Совместно используется всеми фоновыми и пользовательскими поцессами экземпляра.
- Можно настроить через shared_buffers.
- По умолчанию 128 MB.

ИТМО ВТ

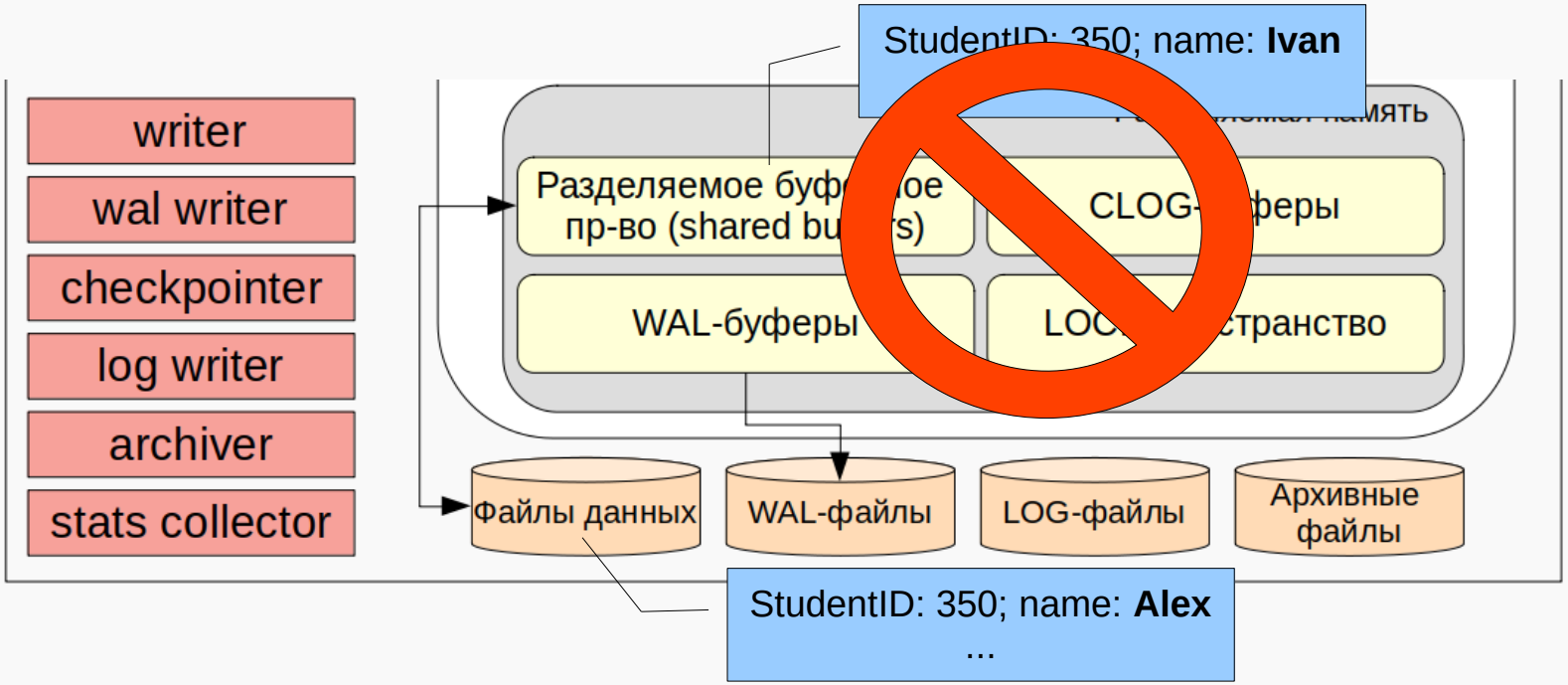
Durability

- Долговечность - при успешном завершении транзакции результаты ее работы должны остаться в системе независимо от возможных сбоев оборудования, системы и тд.
- При этом есть shared buffers — в ОП могут быть измененные данные («грязные» страницы), которые не обновлены в файлах данных.
- Синхронизировать файлы данных при каждом изменении неэффективно.
- Что если сбой произошел до того момента, как измененное содержимое SB попало в файлы данных?

Durability

UPDATE STUDENT

SET Name = 'Ivan' WHERE studentID = 350;



ИТМО ВТ

WAL buffers

- WAL — Write Ahead Log
- Хранит информацию об изменениях данных в БД — записи XLOG.
- Изменениям присваивается LSN — log sequence number.
- Эта информация используется для воссоздания актуального состояния данных в случае восстановления базы данных (например, после сбоя).
- Настраивается через параметр wal_buffers.

ИТМО ВТ

CLOG buffers

- CLOG — Commit Log.
- Хранит данные о статусе проведения транзакций (в процессе, закончена, ...).
- Размер автоматически устанавливается СУБД.
- Доступен серверным процессам.

ИТМО ВТ

Пространство для хранения данных о блокировках (Lock space)

- Хранит данные о блокировках, использующихся экземпляром БД.
- Данные о блокировках доступны всем серверным процессам.
- Можно настроить через max_locks_per_transaction.

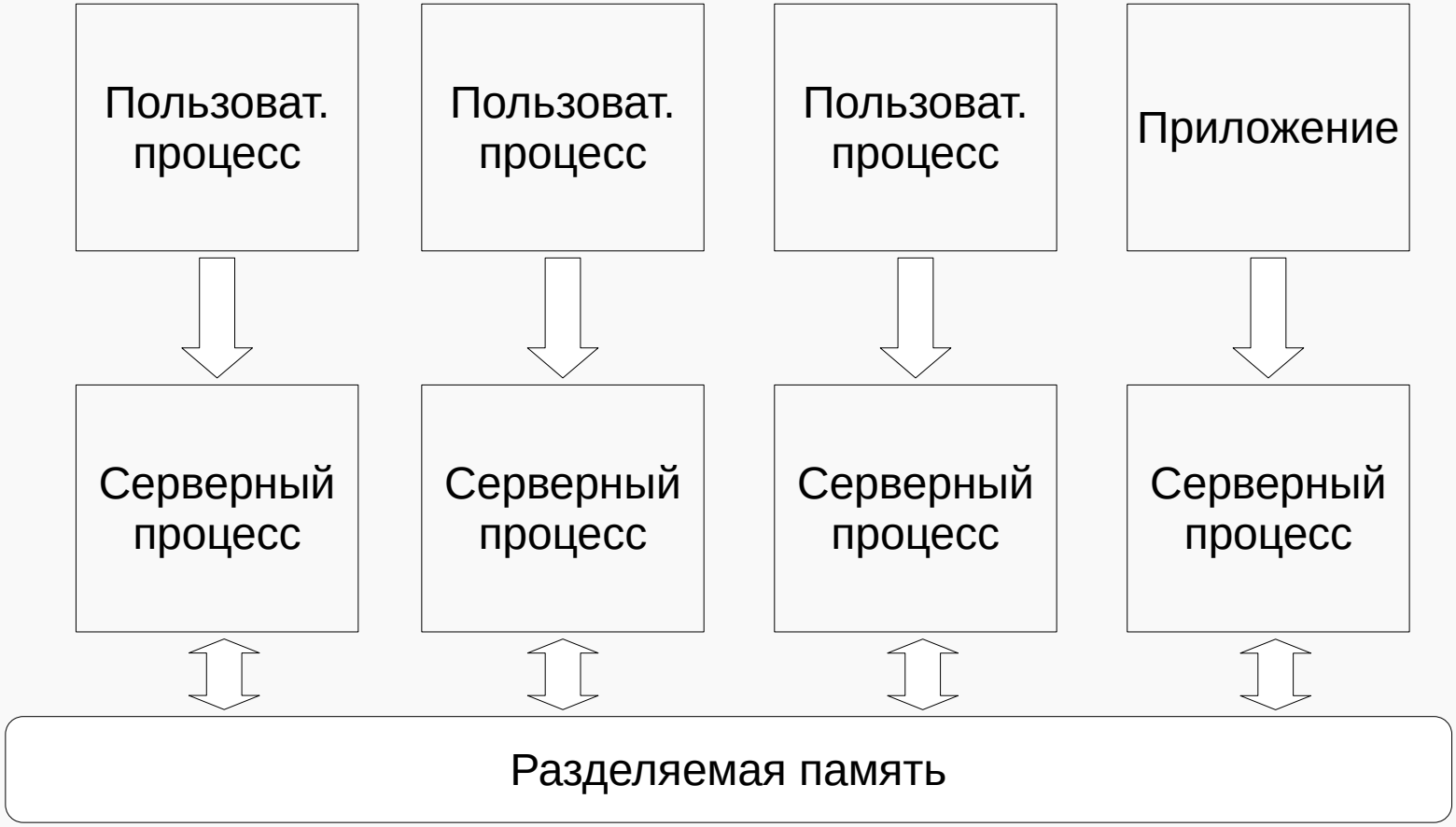
ИТМО ВТ

Буферное пространство процессов БД (неразделяемая область памяти процессов)

- Для каждого серверного пользовательского процесса выделено пространство для осуществления операций.
- По умолчанию — 4 MB.
- Может быть различных видов:
  - vacuum buffers;
  - рабочая память (work_mem) - DISTINCT, ORDER BY, JOIN;
  - вспомогательная рабочая память (maintenance_work_mem) — REINDEX;
  - temp_buffer — для работы со временными таблицами.

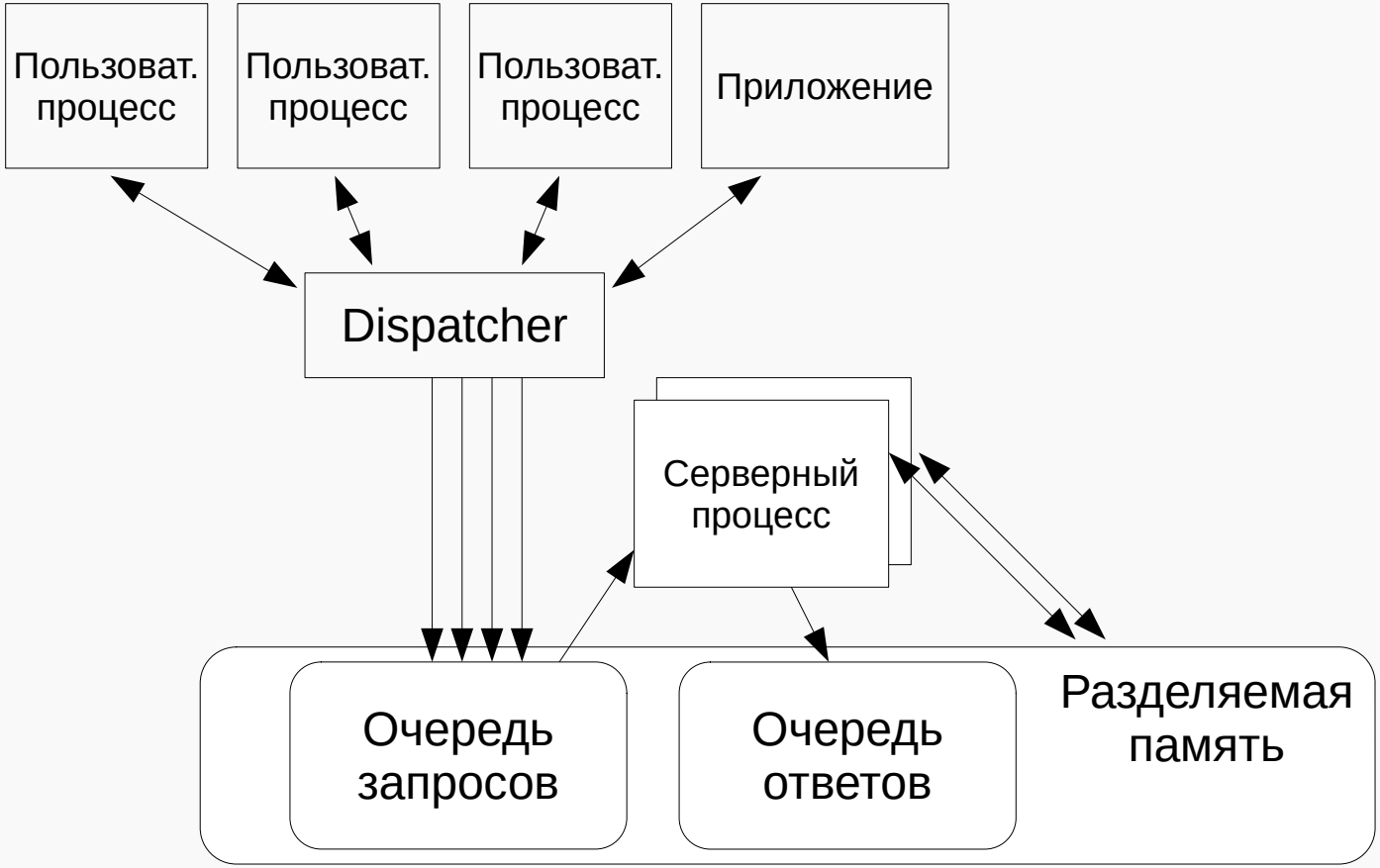
Выделенный сервер (Dedicated Server)

В режиме выделенного сервера каждому пользовательскому процессу создаётся свой «персональный» серверный.



Разделяемый сервер (Dedicated Server)

В режиме разделяемого сервера для каждого пользовательского процесса серверный процесс выделяются диспетчером из специального пула.



ИТМО ВТ

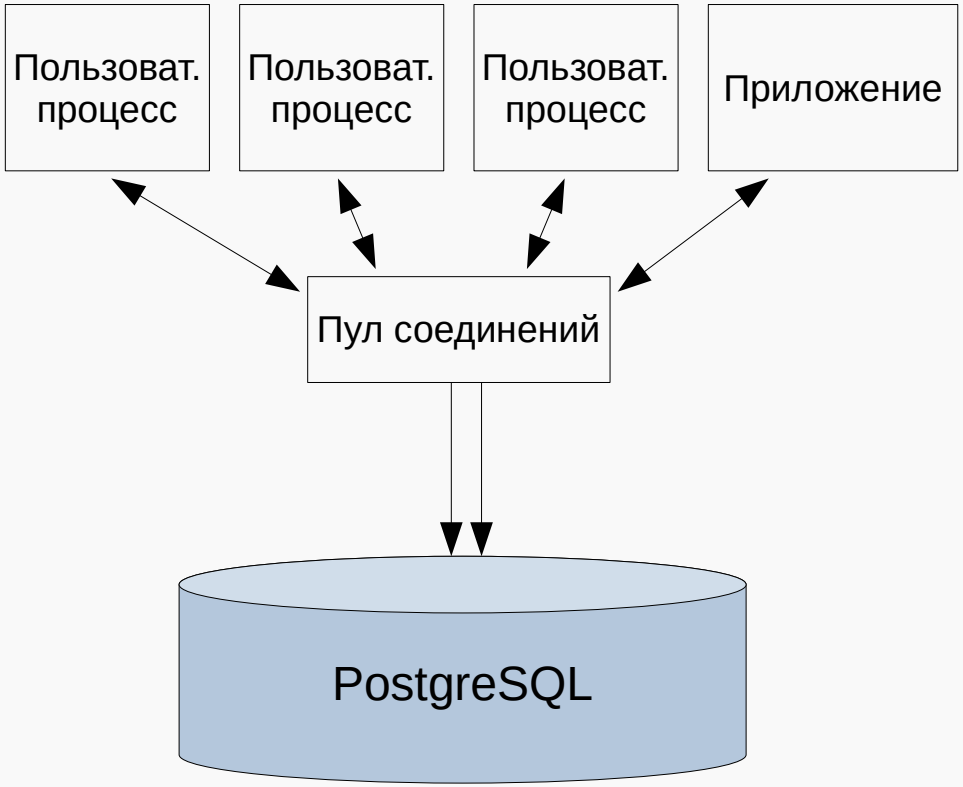
postmaster (postgres)

- Представляет сервер PostgreSQL.
- Отвечает за запуск и остановку PostgreSQL.
- Создает серверные процессы для обработки соединений.
- Управляет данными одного кластера БД.
- Кластер БД — БД или несколько БД, расположенных по определенному адресу в файловой системе.

ИТМО ВТ

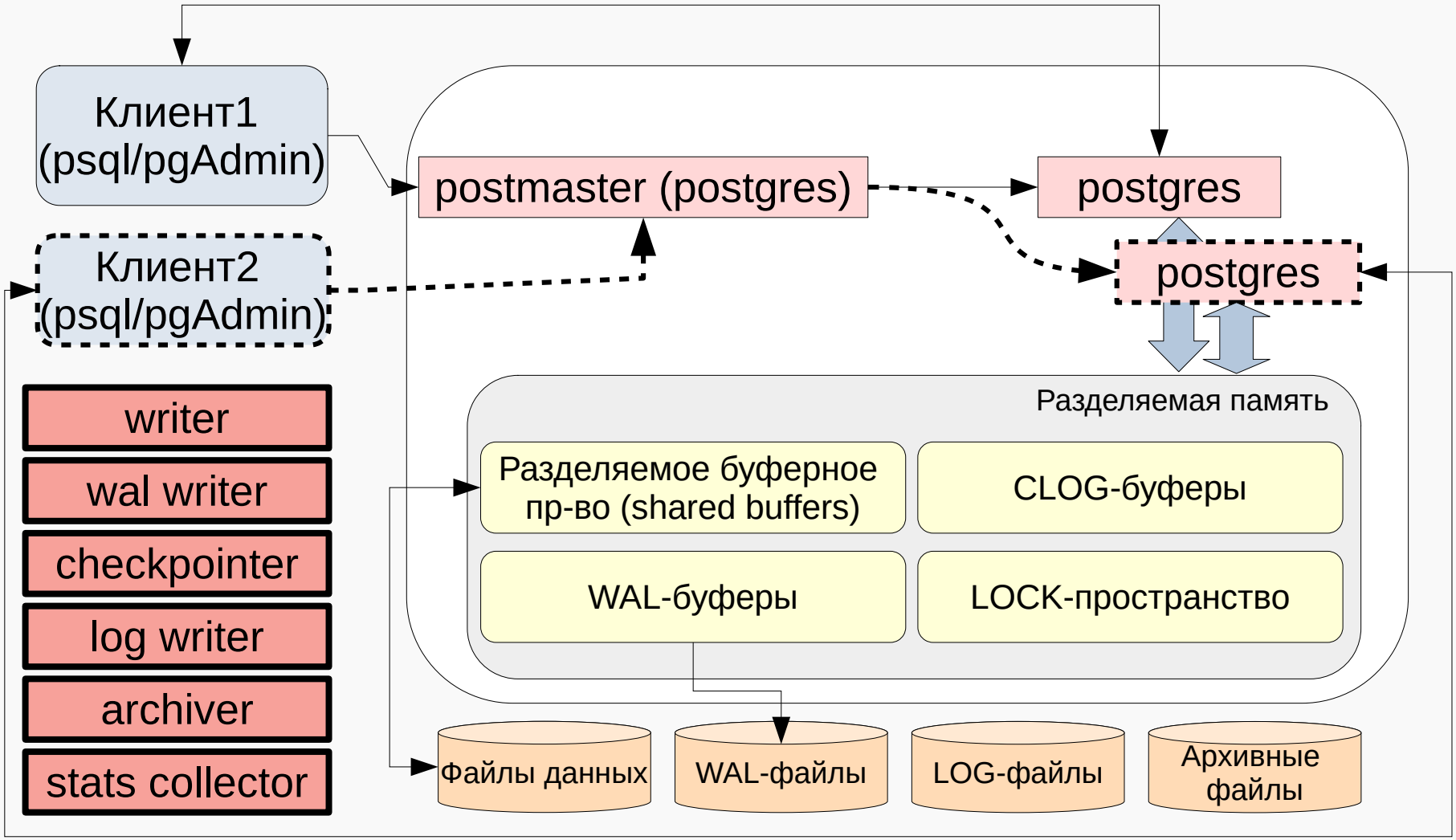
Connection Pool

- pgBouncer
- pgpool-II



ИТМО ВТ

Фоновые процессы PostgreSQL



ИТМО ВТ

Процесс записи в БД
writer process (background writer)

- Периодически записывает измененные (заполненные, «грязные») страницы из SB на диск в файлы данных.
- Помечает записанние страницы «чистыми».
- «Облегчает» работу процесса checkpointer.
- bgwriter_delay

ИТМО ВТ

Работа с контрольными точками checkpointer

- *Контрольная точка (checkpoint)* — точки в последовательности транзакций, в которые произведена синхронизация результатов выполненных операций с файлами на диске.
- Создание контрольной точки:
  - WAL-буфер синхронизируется с диском.
  - «грязные страницы» записываются на диск.
  - Контрольная точка фиксируется в логах.
- Контрольная точка создается, когда заполнен max_wal_size или через время checkpoint_timeout (по умолчанию — 300 секунд) — в зависимости от того, что будет раньше.

logging collector, archiver

logging collector:

- Записывает сообщения, отправленные в stderr, в лог-файлы.
- Для работы нужно, чтобы был установлен параметр logging_collector.

archiver:

- Копирует созданные WAL-файлы в указанное место.
- По умолчанию выключен.

ИТМО ВТ

stats collector

- Служит для сбора различных статистических данных о БД.
- Статистика хранится в промежуточных файлах и через них используется другими процессами.
- Директория с временными файлами определяется параметром stats_temp_directory

- pg_stat_*:

  pg_stat_database, ...

2. Системный каталог

ИТМО ВТ

Системный каталог

- В PostgreSQL есть возможность получения данных о хранимых данных — метаданных:
  - Когда была создана таблица?
  - Сколько в ней атрибутов и какого они типа?
  - Какие индексы связаны с данной таблицей?
- Для доступа к метаданные используются таблицы и представления — **системные каталоги**;
- У каждой БД — есть схема ***pg_catalog***, в ней — каталоги, относящиеся к этой БД.

ИТМО ВТ

Работа с системным каталогом

- Использование системных каталогов напрямую (SELECT * FROM pg_*):

  SELECT count(*) FROM pg_stat_activity;

- Использование INFORMATION_SCHEMA.
- Мета-команды psql.

ИТМО ВТ

Работа с системными каталогами

- **pg_database** — информация о базах данных в кластере БД; создается один для кластера:
- SELECT datdba FROM pg_database WHERE datname = 'MYDB';
- **pg_class** — информация о таблицах, представлениях, индексах и тд.
- **pg_tables** — информация о таблицах текущей БД — у каждого БД свой:

  SELECT * FROM pg_catalog.**pg_tables;**
- Функции: current_user, current_schema, …

ИТМО ВТ

INFORMATION_SCHEMA

- Представления, через которые можно получить данные из системных каталогов.
- Являются частью SQL-стандарта.
- Доступны в PostgreSQL, начиная с версии 8.0.

```
SELECT Table_Name
FROM information_schema.TABLES;
```

ИТМО ВТ

Мета-команды psql

- Посмотреть структуру таблицы:

  \d MyTable
- Информация об объекте базы данных:

  \d objectName
- Посмотреть созданные индексы:

  \di
- ...

ИТМО ВТ

oid - идентификатор объекта в PostgreSQL

pg_attribute - каталог с информацией о колонках таблицы.

Некоторые поля:

- attrelid - oid таблицы, к которой относится эта колонка (из каталога pg_class);
- attname - имя колонки.

Для типа oid существуют различные **типы-алиасы** для упрощения работы (например, **regclass** ~ oid таблицы, позволяет получить значение oid по названию таблицы).

ИТМО ВТ

Пример

Если в БД одна таблица STUDENT, название колонок этой таблицы:

```
SELECT attname FROM pg_attribute
WHERE attrelid = (SELECT oid FROM pg_class
WHERE relname = 'STUDENT');
```

```
SELECT attname FROM pg_attribute WHERE attrelid =
'STUDENT'::regclass;
```

ИТМО ВТ

Пример работы с pg_tables

SELECT * FROM pg_catalog.**pg_tables;**

Будет ли работать такой запрос?

SELECT * FROM **pg_tables;**

ИТМО ВТ

Схема

- Используются для логической группировки объектов в БД.
- У каждой БД в PostgreSQL есть схема public.
- Полное имя в PostgreSQL:

```
dbName.schemaName.objectName
```

```
dbName.schemaName1.objName
dbName.schemaName2.objName
```

} Разные объекты

ИТМО ВТ

search_path

- Объекты можно использовать без полного имени — тогда используется search_path.
- search_path — последовательность схем, которая будет использована для идентификации объекта, когда используется неполное имя.
- Схемы рассматриваются в порядке из search_path.
- Первая схема из search_path — используется для создания объектов — текущая.

ИТМО ВТ

search_path

- pg_temp и pg_catalog автоматически добавляются в search_path перед первой указанной схемой (порядок можно переопределить).
- Изменять search_path нужно осторожно — меняется контекст выполнения запросов (разрешение имен).
- show search_path

3. Управление доступом к БД

ИТМО ВТ

Пользовательские права

- Разным категориям пользователей должны предоставляться:
  - разные возможности (в зависимости от их потребностей);
  - для управления различных объектов БД.

- Возможности — обеспечение доступа (или выполнения другой операции) с таблицами, представлениями; создание пользователей.

ИТМО ВТ

Привилегии

- Предоставляемые возможности определяются **привилегиями**;
- Привилегии:
  - системные — описывают возможность осуществления операций над БД;
  - для взаимодействия с объектами — операции над различными объектами (контроль над данными в объектах БД);
    - с различными объектами связаны различные привилегии.

ИТМО ВТ

Работа с привилегиями

GRANT privilegeName1, privilegeName2, …

[ON table1]

TO user1, user2, user3;

GRANT CREATE ON SCHEMA someSch TO sXXXXXX;

- Привилегии: SELECT, INSERT, UPDATE, DELETE, CREATE, EXECUTE, CONNECT, REFERENCES, …
- ALL PRIVILEGES — для выдачи всех привилегий (в зависимости от контекста).

ИТМО ВТ

Владелец объекта

- **Владелец объекта** – обычно пользователь (роль), создавший объект.
- Обладает некоторыми привилегиями для созданного объекта по умолчанию, например:
  - ALTER
  - DROP

ИТМО ВТ

Роли

- Привилегии отражают конкретную возможность.
- Роли — именованные наборы привилегий:
  - позволяют управлять объектами и БД на более «высоком» уровне;
  - могут выступать в качестве пользователей;
- Роль - конфигурируется на уровне кластера:
  - назначение и привилегии могут отличаться для разных БД;
  - Имя роли — уникально для кластера.

ИТМО ВТ

Роли

- **CREATE ROLE**:
  - CREATE ROLE STUDENT;
  - CREATE ROLE STUDENT **WITH LOGIN** PASSWORD 'somePwd';
  - CREATE ROLE STUDADMIN **CREATEROLE;**
- **ALTER ROLE**
- **DROP ROLE**

- Могут использовать суперпользователи, а также роли с CREATEROLE (на не суперпользователях).

ИТМО ВТ

Пользователи

- При установке кластера — создается администратор (postgres).
- Созданы для отображения именованных пользователей системы.
- Могут быть созданы суперпользователем и пользователем с ролью CREATEROLE.

- **CREATE USER**:
  - CREATE USER s234XXX WITH PASSWORD 'somePwd'**;**

ИТМО ВТ

Пользователи

- Создание пользователя аналогично созданию роли с параметром LOGIN:

  CREATE USER s234XXX;

  CREATE ROLE s234XXX WITH LOGIN;

- При создании БД создается роль public;
  - public назначается всем пользователям и ролям — определяет права пользователей и ролей по умолчанию к разным объектам.

- pg_roles

ИТМО ВТ

Настройка ролей

- **SUPERUSER** — пользователь может действовать как суперпользователь кластера (все права на все объекты);
  - можно создать нескольких суперпользователей
- **NOSUPERUSER** — убирает возможности SUPERUSER.

- **CREATEROLE** — позволяет создавать роли.
- **CREATEDB** — позволяет создавать базы в кластере.

ИТМО ВТ

Настройка ролей (2)

- **PASSWORD** - установить пароль:
  - CREATE USER s234XXX WITH PASSWORD 'somePwd'**;**
- **PASSWORD NULL** - запретить пользователю вход по паролю.
- **CONNECTION LIMIT n** — задать ограничение по числу подключений для пользователя.
- **VALID UNTIL** — установить срок действия роли (срок действия пароля этой роли).

ИТМО ВТ

Подключение к БД

Для подключения к БД роль должна быть:

- LOGIN;
- содержать привилегию CONNECT на нужную БД;
- разрешение в pg_hba.conf;

ИТМО ВТ

Группы ролей

- Роли можно объединять в группы для более гибкого управления ими:

  CREATE ROLE s123456 WITH LOGIN PASSWORD 'somsdfsld';

  CREATE ROLE STUDENTS;

  GRANT STUDENTS TO s123456;

  CREATE ROLE s123457 WITH LOGIN PASSWORD 'xfsewrew' IN ROLE students;

ИТМО ВТ

Админ группы ролей

- Участник группы (или несколько участников группы) могут быть администраторами:

  CREATE ROLE students WITH NOLOGIN ADMIN s123456;

- Или с помощью GRANT:

  GRANT students TO s123456 WITH ADMIN OPTION;

- Роль-администратор может добавлять новых участников в группу (как и суперпользователь кластера).
- Роль должна быть создана, чтобы быть добавлена в качестве администратора.

ИТМО ВТ

Работа с ролями

- WITH GRANT OPTION — указывается возможность дальнейшей передачи роли от того, кому она назначена:

  GRANT UPDATE ON STUDENT TO s458455 WITH GRANT OPTION;

- INHERIT/NOINHERIT — наследование привилегий при работе с группами.
- SET ROLE

- Удаление группы не удаляет ее участников.

ИТМО ВТ

INHERIT

```
CREATE ROLE s123456 WITH LOGIN PASSWORD
'somsdfsld';
```

Создали пользователя
для студента

```
REVOKE ALL ON stud_comments FROM s123456;
```

Убираем все права
на таблицу stud_comments

```
INSERT INTO stud_comments VALUES ...;
```

Ошибка: нет нужных прав

ИТМО ВТ

INHERIT

```
CREATE ROLE STUDENTS WITH NOLOGIN;
```

Создали группу ролей для студентов

```
GRANT INSERT ON stud_comments TO STUDENTS;
```

Выдали права студентам на изменение таблицы

ИТМО ВТ

INHERIT

GRANT STUDENTS TO s123456;

Добавляем s123456
в группу

psql -U s123456 ucheb

INSERT INTO stud_comments VALUES ...;

Значения добавлены

ИТМО ВТ

INHERIT

- Участники группы получают права ролей-групп их окружающих.
- INHERIT по умолчанию.

CREATE ROLE s123456 WITH LOGIN PASSWORD 'somsdfsld';

CREATE ROLE s123456 WITH LOGIN PASSWORD 'somsdfsld' **INHERIT**;

ИТМО ВТ

NOINHERIT

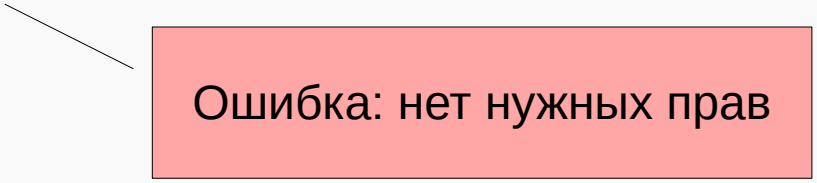
CREATE ROLE s123457 WITH LOGIN PASSWORD 'somsdfsld' **NOINHERIT**;

GRANT INSERT ON stud_comments TO STUDENTS;

GRANT STUDENTS TO s123457;

psql -U s123457 ucheb

INSERT INTO stud_comments VALUES ...;



SET ROLE

psql -U s123457 ucheb

SET ROLE TO STUDENTS;

INSERT INTO stud_comments VALUES ...;

Значения добавлены

ИТМО ВТ

Поиск привилегий

1) Поиск среди привилегий роли.
2) Поиск среди родителей (если у изначальной роли INHERIT). Поиск среди прародителей (если у родителей INHERIT).
3) Есть ли привилегия для роли public.

ИТМО ВТ

Отмена привилегий

REVOKE privilegeName1, privilegeName2, …

[ON table1]

TO user1, user2, user3;

REVOKE UPDATE, DELETE, TRUNCATE ON STUDENT FROM s4343453;

ИТМО ВТ

Документация

Документация PostgreSQL:
https://www.postgresql.org/docs/14/index.html

Лицензия PostgreSQL: