Технологии и разработка СУБД

Лекция 2. Обзор архитектуры

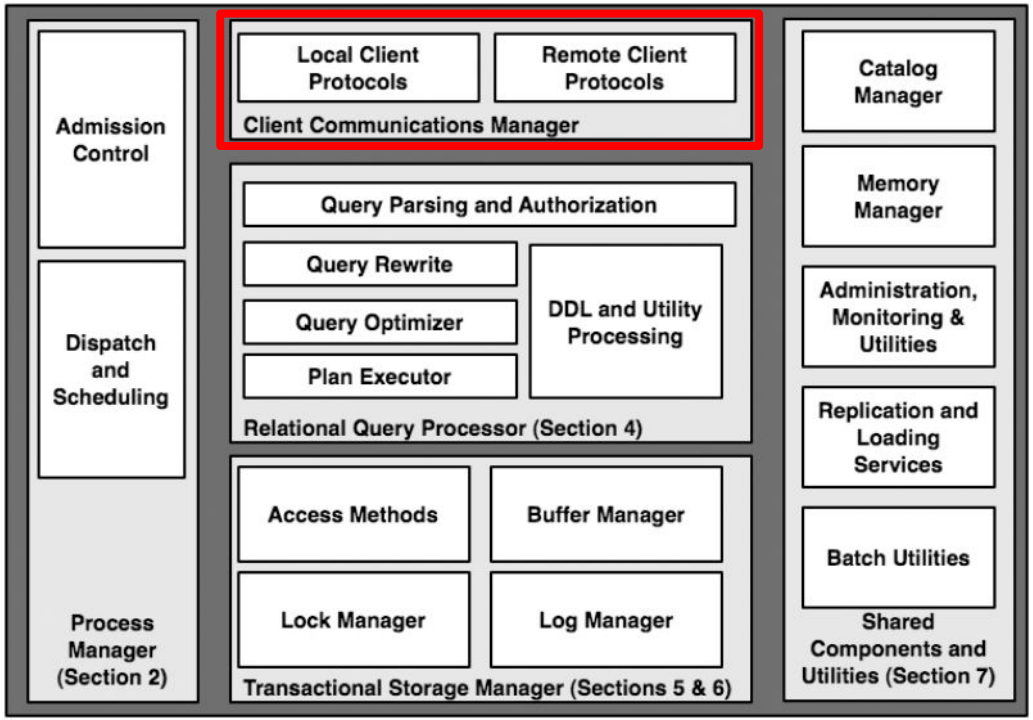
Анастасия Лубенникова
Александр Алексеев

Лекция 2

- Часть 1: Общая архитектура РСУБД
- Часть 2: Модели процессов

Часть 1: Общая архитектура РСУБД

Основные компоненты



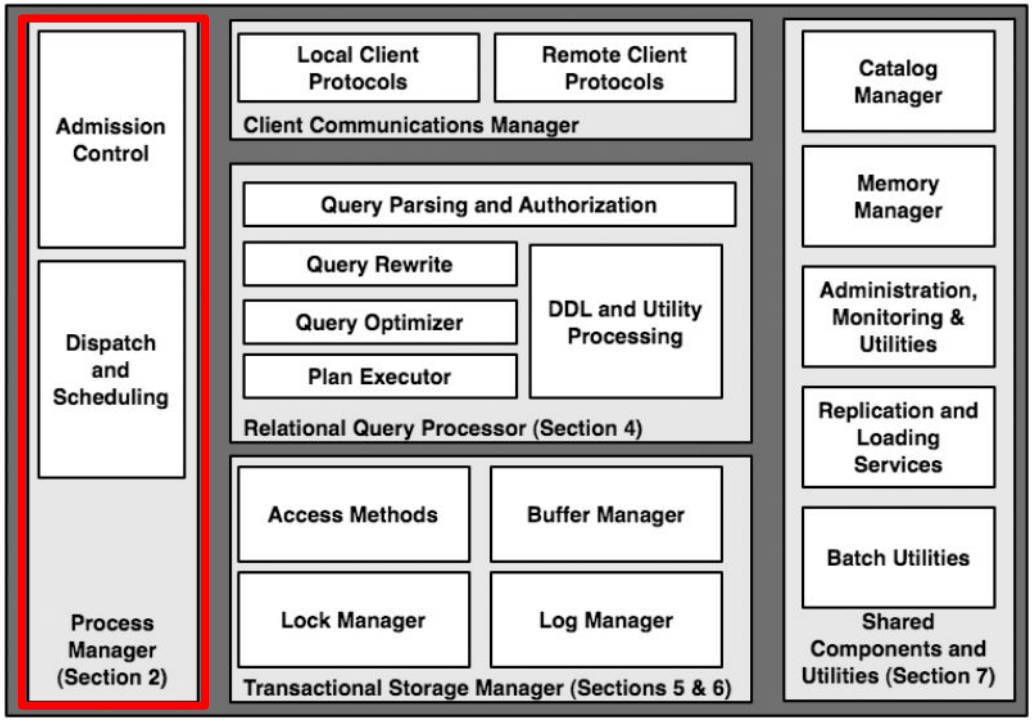
Client communication manager

- клиент-серверная архитектура
- основные действия:
  - устанавливать соединение, запоминать состояние
  - принимать SQL команды
  - возвращать данные
  - возвращать управляющие сообщение (коды ответа, коды ошибок)
- различные протоколы

Client communication manager. PostgreSQL

- https://postgrespro.ru/docs/postgrespro/9.6/protocol
- libpq
  - библиотека на C
  - основной клиентский интерфейс
  - входит в стандартную поставку постгреса
- C++, Perl, Python, Tcl, ODBC ...
- JDBC (Java Database Connectivity)

Основные компоненты



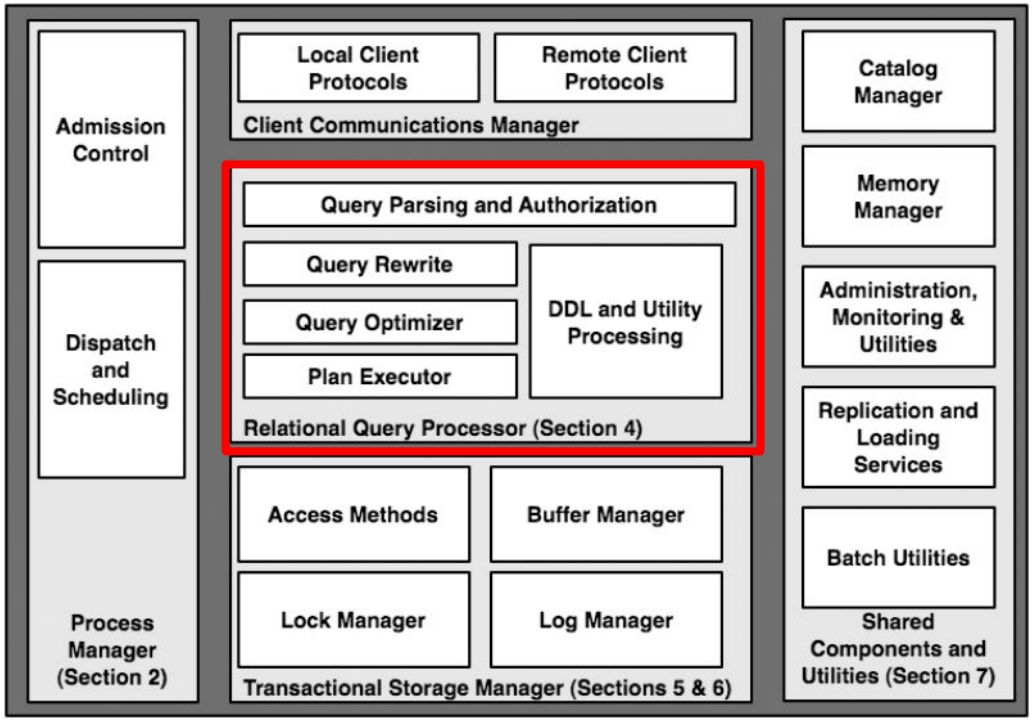
Process Manager

- создание “воркера” для обслуживания подключенного клиента
- управление служебными “воркерами”
- контроль доступа
  - права на подключение
  - доступность необходимых ресурсов
  - приоритизация запросов

Process Manager. PostgreSQL

- процесс postgres (postmaster)
  - создает отдельный процесс для каждого подключения
  - управляет всеми служебными процессами
- контроль доступа
  - файл конфигурации: pg_hba.conf

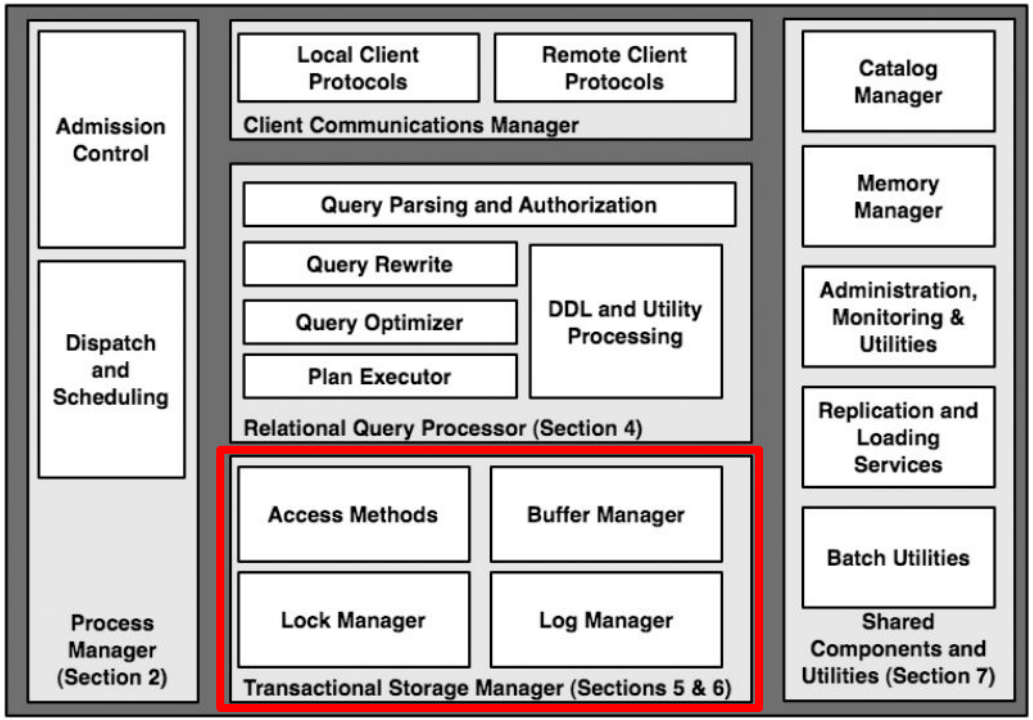
Основные компоненты



Query processor

- Парсер (Parser)
- Авторизация пользователя
  - проверка прав доступа к данным
- выполнение DDL запроса
  - DDL and Utility processor
- выполнение DML запроса
  - Планировщик (Planner)
  - Оптимизатор (Optimizer)
  - Исполнитель (Executor)

Основные компоненты



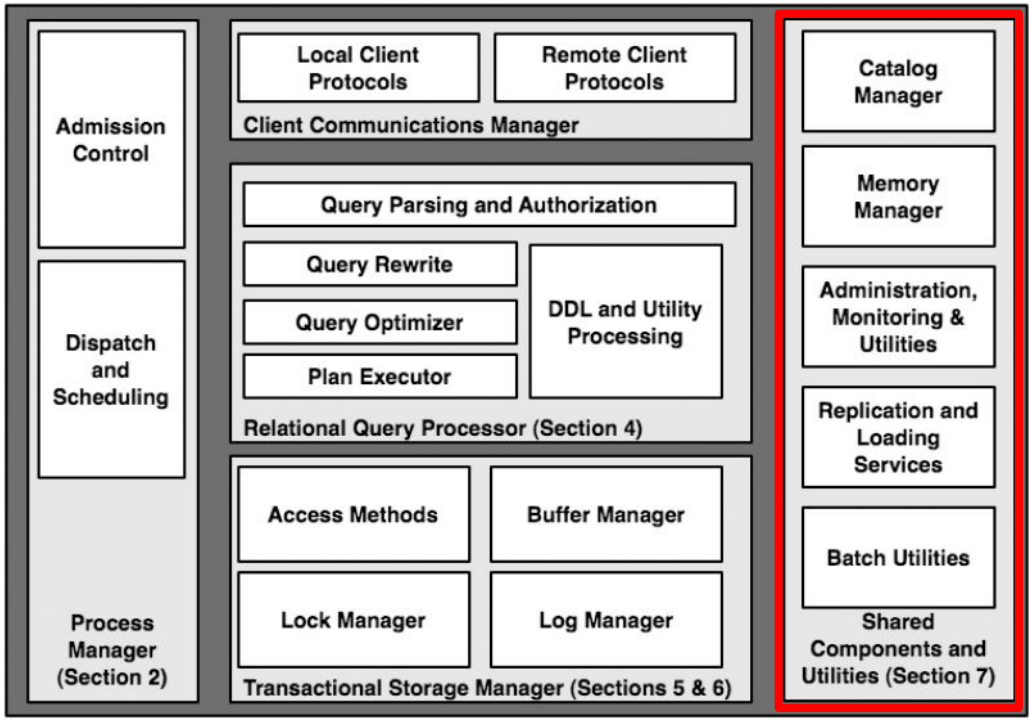
Transactional Storage Manager

- Методы доступа (Access methods)
  - структуры данных и алгоритмы работы с данными на диске
  - физическое расположение данных
  - таблицы / индексы
- Разделяемый кэш страниц (Buffer Manager)
  - для ускорения доступа к данным
- Управление блокировками
  - для обеспечения конкурентного доступа к страницам данных
- Управление логом транзакций
  - для выполнения свойств ACID (атомарность и надежность)
- MVCC

Ответ на запрос

- Ответ возвращается клиенту через Client Communication Manager

Основные компоненты



Shared Components and Utilities

- Системный каталог
  - хранит информацию об объектах базы (таблицы, права доступа, зависимости...)
- Управление памятью
- Управление логами
- Управление настройками
- Сборка статистики

Часть 2: Модели процессов

Модели многопоточности

- Process
  - Не разделяемая память
  - Взаимодействие с помощью сигналов, пайпов, сокетов, ...
- Thread
  - Разделяемая память
  - Взаимодействие с помощью мьютексов, атомарных переменных, TLS, ...
- Корутины / легковесные потоки
  - Выглядят как потоки, но мапятся на потоки операционной системы N:M (N > M)
  - Дешевле переключение контекста, асинхронный I/O
  - В языке (Go, Haskell) или в виде библиотеки (Scala, Rust)
- Акторы
  - Легковесные потоки без разделяемой памяти
  - Взаимодействие с помощью сообщений, линков и мониторов
  - Примеры: Erlang, Scala (Akka), Cloud Haskell

Модели многопоточности

DBMS Worker - кусок кода, обслуживающий подключение клиента.

- OS Process per Worker
  - PostgreSQL, IBM DB2, Oracle
- OS Thread per Worker
  - IBM DB2, Microsoft SQL Server, MySQL
- Process pool
  - PgBouncer for PostgreSQL, DRCP for Oracle, …
- Coroutine per Worker
  - RethinkDB, а также CockroachDB, и вообще мир Go
- Actor per Worker
  - Riak, Couchbase, и вообще мир Erlang

Домашнее задание №1. Все получилось?

- Настроить Linux-окружение для разработки (можно виртуалку)
- Освежить знания C и Git
- Собрать PostgreSQL из исходников и установить
- Спроектировать схему базы данных “телефонная книга”

Домашнее задание №2

- Выберите домашнее задание на семестр и сроки его сдачи

Вариант 1

- ZSON: использовать PGLZ напрямую, не полагаясь на эвристики PostgreSQL
- С обратной совместимостью!
- https://github.com/postgrespro/zson
- https://habr.ru/p/312006/
- https://eax.me/postgresql-extensions/

Вариант 2

- Задокументировать хуки / написать о них статью
- Был старый доклад: https://wiki.postgresql.org/images/e/e3/Hooks_in_postgresql.pdf

Вариант 3

- Расширение для кодирования/декодирования QTH Locator
- https://ru.wikipedia.org/wiki/QTH-%D0%BB%D0%BE%D0%BA%D0%B0%D1%82%D0%BE%D1%80

Вариант 4

- Сделать расширение-хук “пользователь готовится авторизоваться”, “пользователь неуспешно авторизовался”, “пользователь успешно авторизовался” и тд
- Хотя бы банально для логирования.

Вариант 5

- pg_protobuf: добавить возможность модифицировать данные
- https://github.com/afiskon/pg_protobuf
- https://eax.me/cpp-protobuf/

Вариант 6

- pg_protobuf: написать генератор PL/pgSQL-процедур для доступа к полям Protobuf из .proto файлов
- https://github.com/afiskon/pg_protobuf
- https://eax.me/cpp-protobuf/
- http://shop.oreilly.com/product/9780596155988.do

Вариант 7

- wal2json: добавить возможность фильтрации таблиц
- https://github.com/eulerto/wal2json
- https://eax.me/postgresql-logical-decoding/

Вариант 8

- Расширение для просмотра списка загруженых .so/DLL
- SELECT list_shared_libraries();
- Как минимум совместимость с Linux и Windows, желательно также MacOS и FreeBSD.

Вариант 9

- Расширение для просмотра списка открытых файловых десрипторов и что это: сокет, файл, unix socket, pipe, ...

Вариант 10

- Прочитать пейпер, понять, рассказать классу на семинаре
- Amazon Aurora: Design Considerations for High Throughput Cloud - Native Relational Databases
- Caribou: Intelligent Distributed Storage (FPGA based DBMS)
- Clock-SI: Snapshot Isolation for Partitioned Data Stores Using Loosely Synchronized Clocks
- Dynamo: Amazon’s Highly Available Key-value Store
- In Search of an Understandable Consensus Algorithm, Extended Version (Raft paper)
- CASPaxos: Replicated State Machines without logs

Вариант 10 (продолжение)

- Large-scale Incremental Processing Using Distributed Transactions and Notifications (Percolator paper)
- Spanner: Google’s Globally-Distributed Database
- Spanner: Becoming a SQL System
- What’s Really New with NewSQL?
- A Critique of ANSI SQL Isolation Levels

Вариант 11

- HurmaDB: оптимизировать выполнение range-запросов для больших диапазонов (не строить весь ответ в памяти)
- https://github.com/afiskon/hurmadb

Вариант 12

- HurmaDB: рефакторинг, перейти с pthreads на std::thread
- https://github.com/afiskon/hurmadb
- https://eax.me/cpp-multithreading/

Дополнительные материалы

- The Morning Paper https://blog.acolyer.org/
- Databass Papers https://t.me/databasss

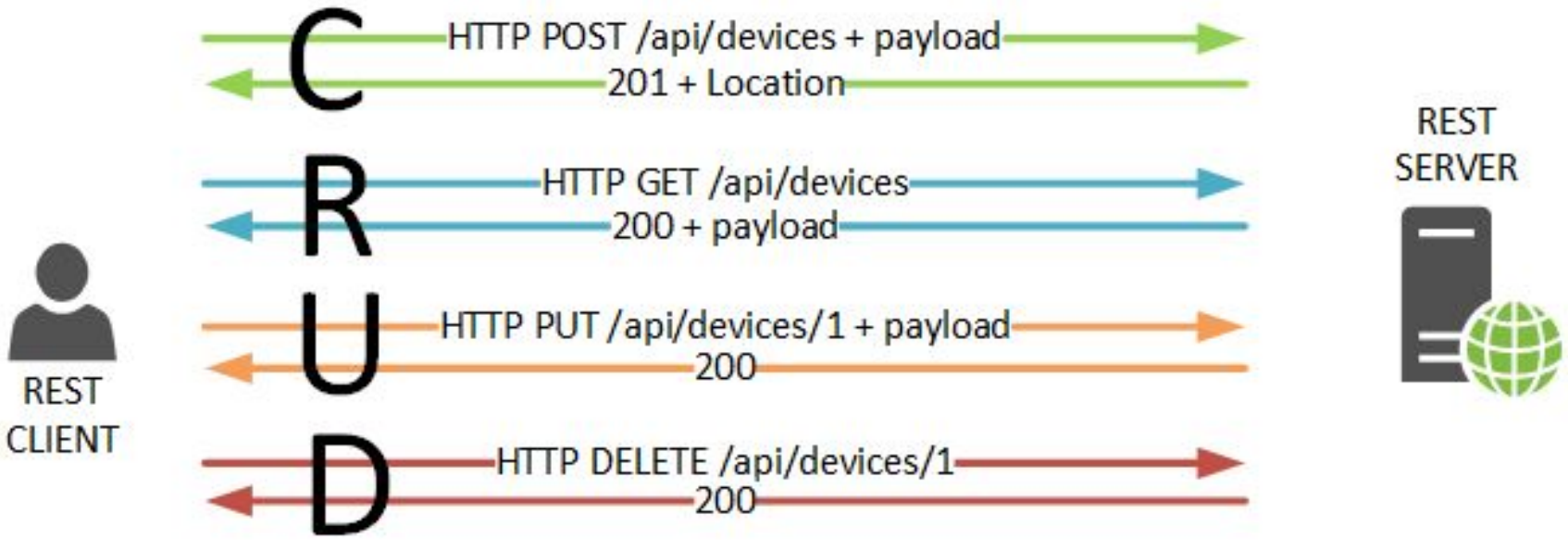
Вопросы и ответы.

- a.lubennikova@postgrespro.ru
- a.alekseev@postgrespro.ru
- Telegram: https://t.me/dbmsdev

Бонус-слайды. Вопросы залу

- Что вы знаете о процессе разработки, Agile / XP / Scrum, видах тестирования, фич-бранчах, code review, CI, и т.д?
- Нужно ли объяснить, что представляет собой HTTP?

Бонус-слайды. REST API



Бонус-слайды. Протокол Memcached

>>> version

<<< VERSION 1.4.10

>>> get example_key

<<< VALUE example_key 0 3

<<< 128

<<< END

>>> decr example_key 3

<<< 125