Метод параллельного выполнения запросов к системе управления базами данных PostgreSQL в пределах одного соединения

Студент: Платонова Ольга Сергеевна

Группа: ИУ7-85Б

Руководитель: Филиппов Михаил Владимирович,
к.т.н., доцент кафедры ИУ-7

Консультант: Гаврилова Юлия Михайловна

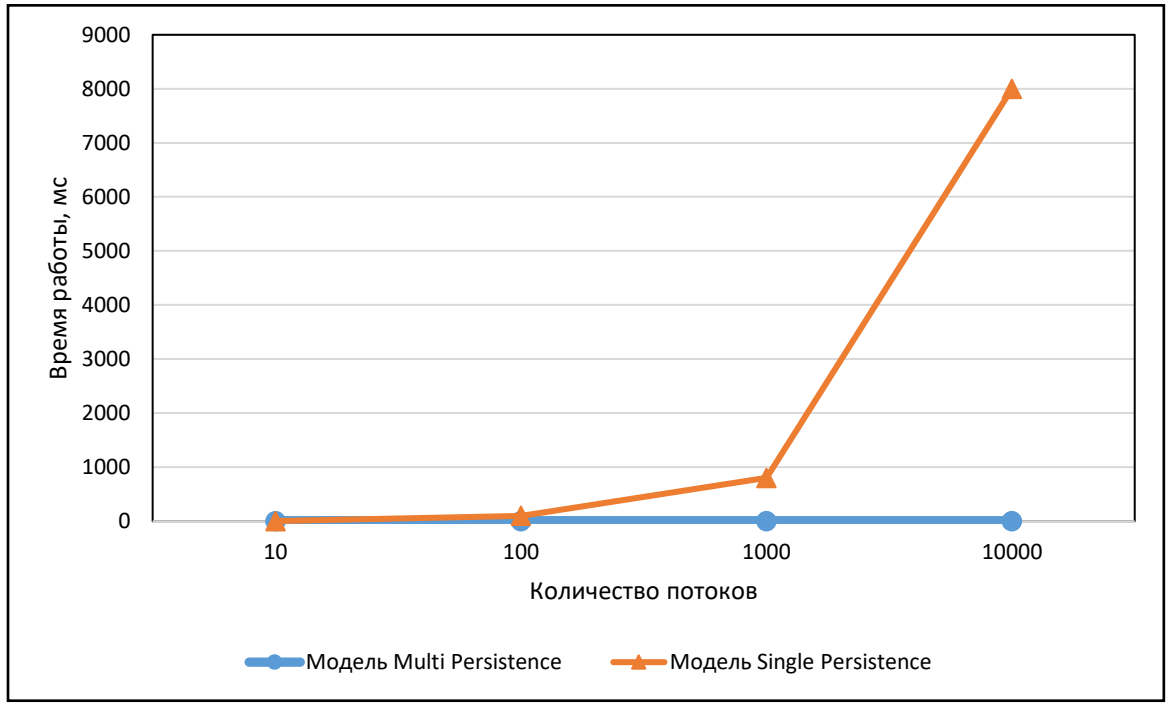
Цель и задачи работы

Цель – разработать метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

Задачи:

- Выполнить анализ предметной области и существующих методов выполнения запросов в MPP системах;
- Разработать метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения;
- Реализовать разработанный метод;
- Выполнить сравнительный анализ времени работы метода и различных реализаций выполнения запросов.

Однопоточное и многопоточное соединение с базой данных



Доступ к БД объемом 100.000 записей.
Многопоточная программа:

- в 1000 раз быстрее;
- ошибка памяти при 10.000 и более соединений.

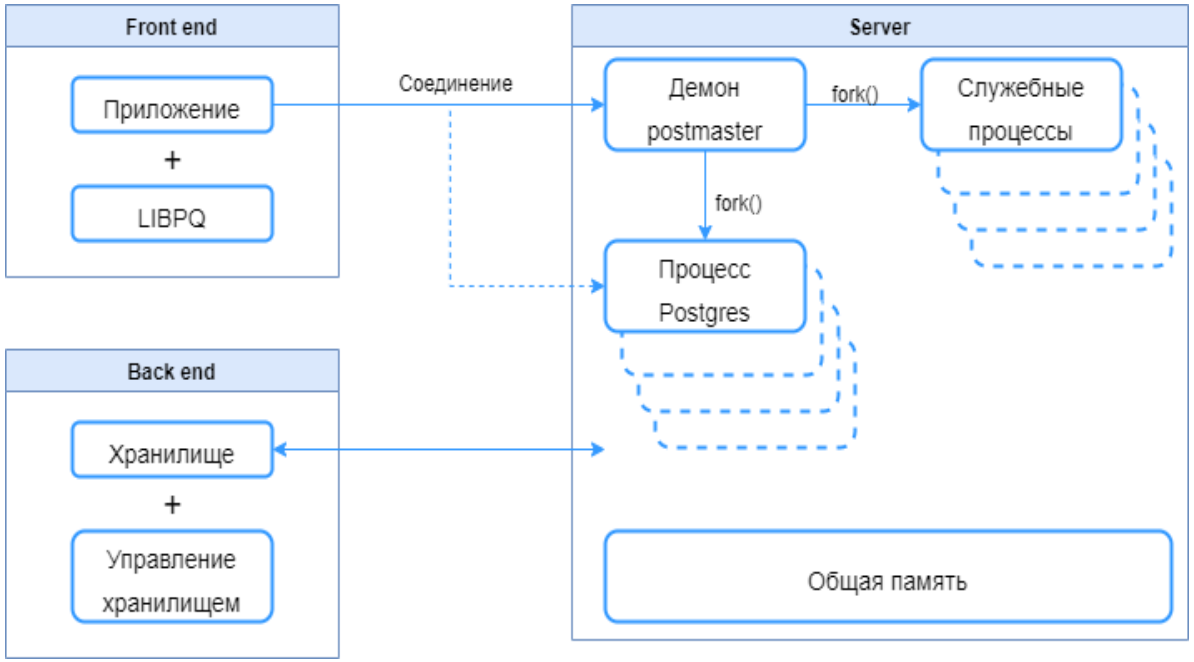
Процесс подключения занимает
2-3 МБ памяти.

Создание соединения в PostgreSQL

Рейтинг	СУБД	Модель БД
1.	Oracle	Реляционная
2.	MySQL	Реляционная
3.	Microsoft SQL Server	Реляционная
4.	**PostgreSQL**	**Реляционная**
5.	MongoDB	Документная

PostgreSQL (14.2):

- доступность исходного кода;
- кроссплатформенность.



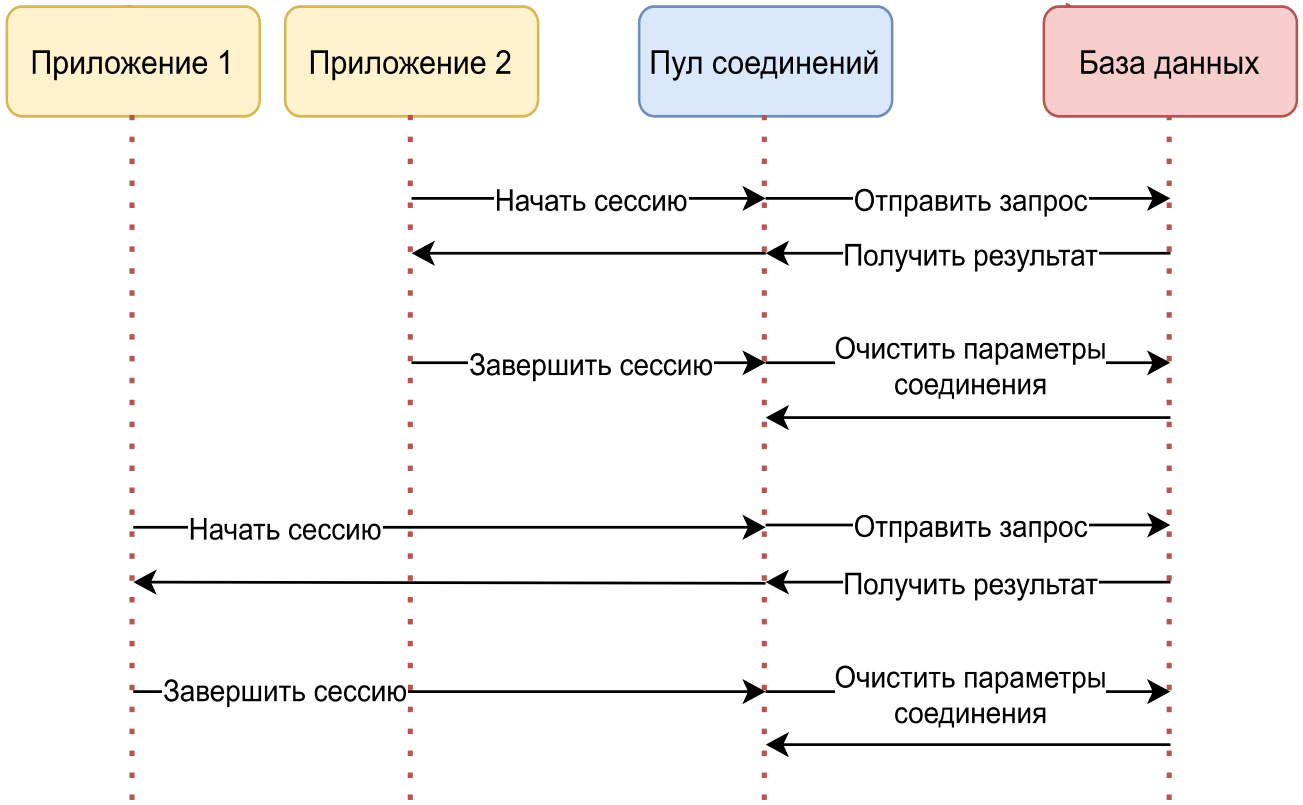
Архитектура PostgreSQL

Пул соединений

Пул соединений — набор открытых и готовых к использованию соединений с базой данных.

Особенно заметно повышение производительности, когда размер пула не превосходит числа потоков.

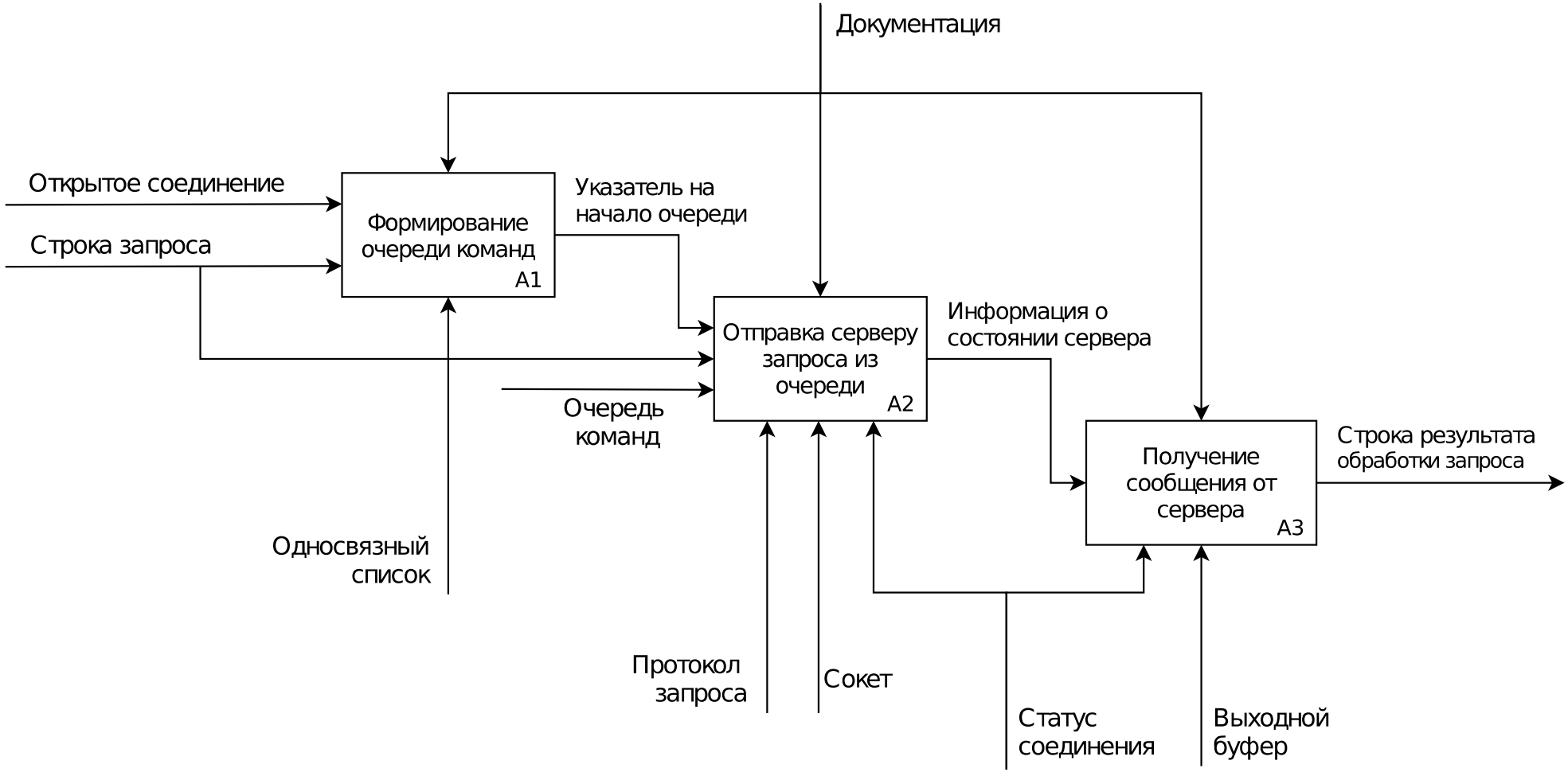
- **Встроенный пул**
- **Пул на основе libpq**
- **Пул в качестве внешней службы**



Решения на основе пула соединений

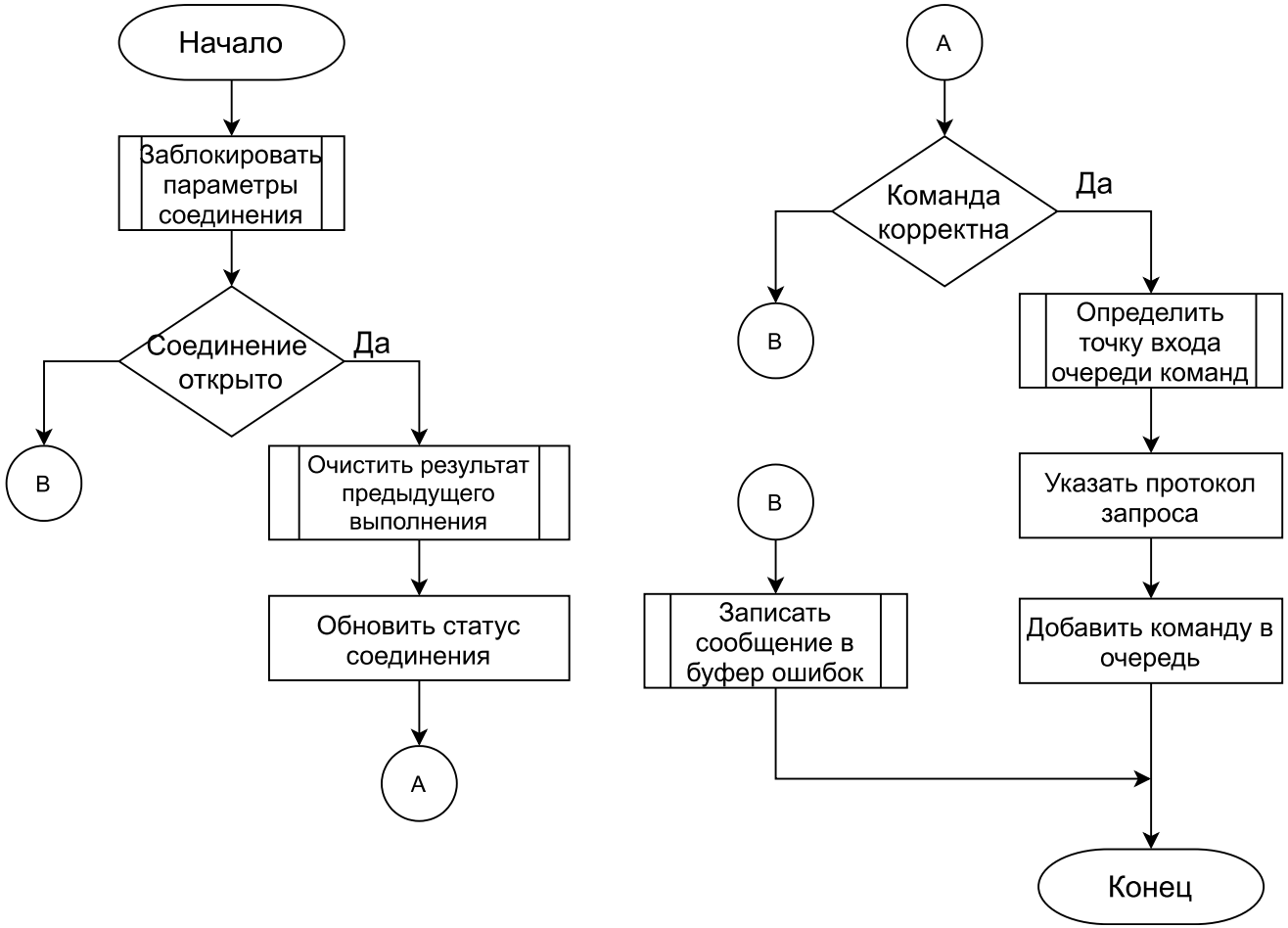
Вид пула / Критерий	Пул на основе libpq	Пул в качестве внешней службы	Встроенный пул
Максимальный размер пула по умолчанию	100	100	32767
Наличие лицензии	Нет	Нет	Да
Необходимость поддержки отдельного пула для каждой БД	Да	Да	Да
Необходимость самостоятельной разработки и поддержки	Да	Нет	Нет

Функциональная диаграмма разрабатываемого метода



Формирование очереди команд и отправка серверу запроса из очереди

Алгоритм формирования очереди команд



Процесс отправки запроса серверу

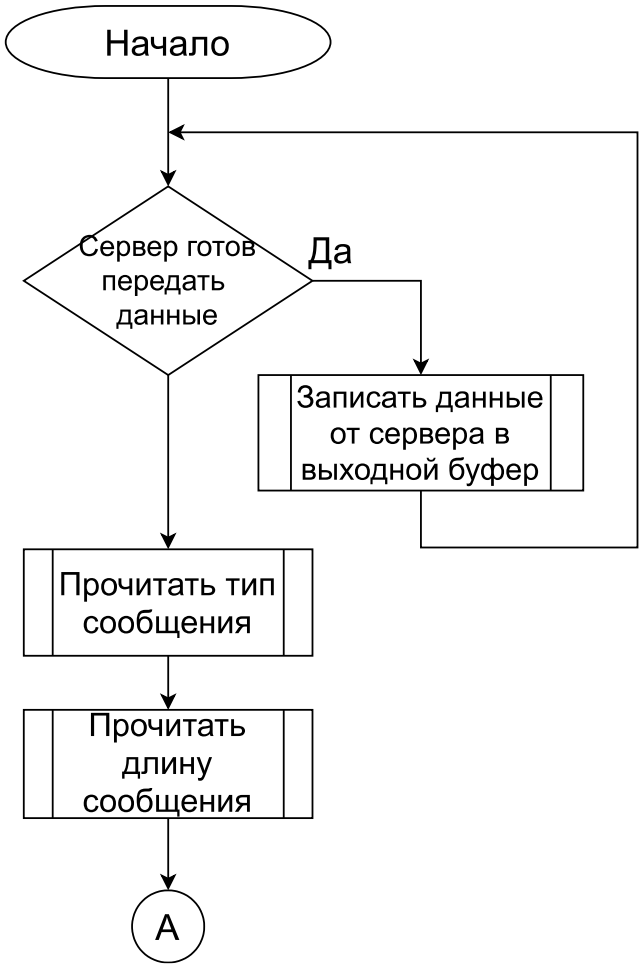
Для каждого соединения определен выходной буфер данных, которые еще не были отправлены на сервер.

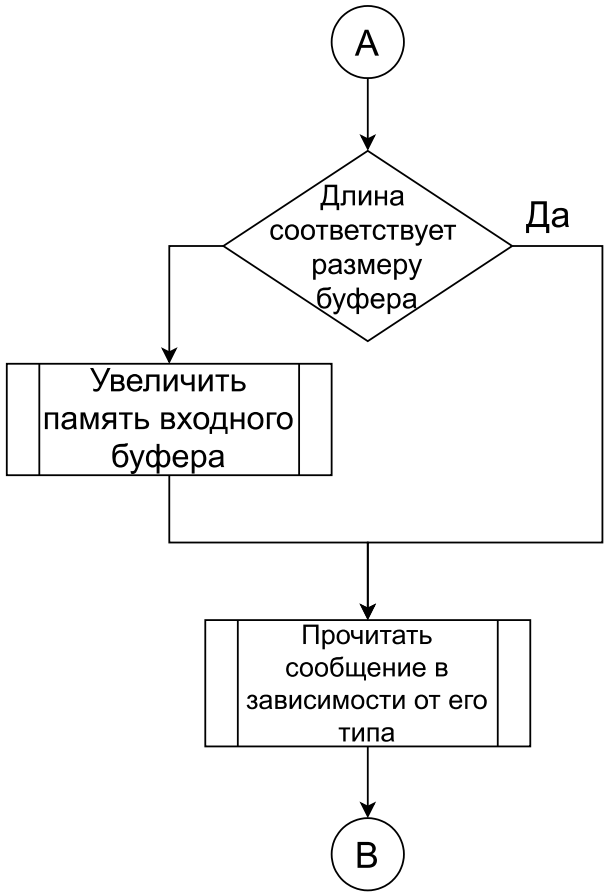
Query

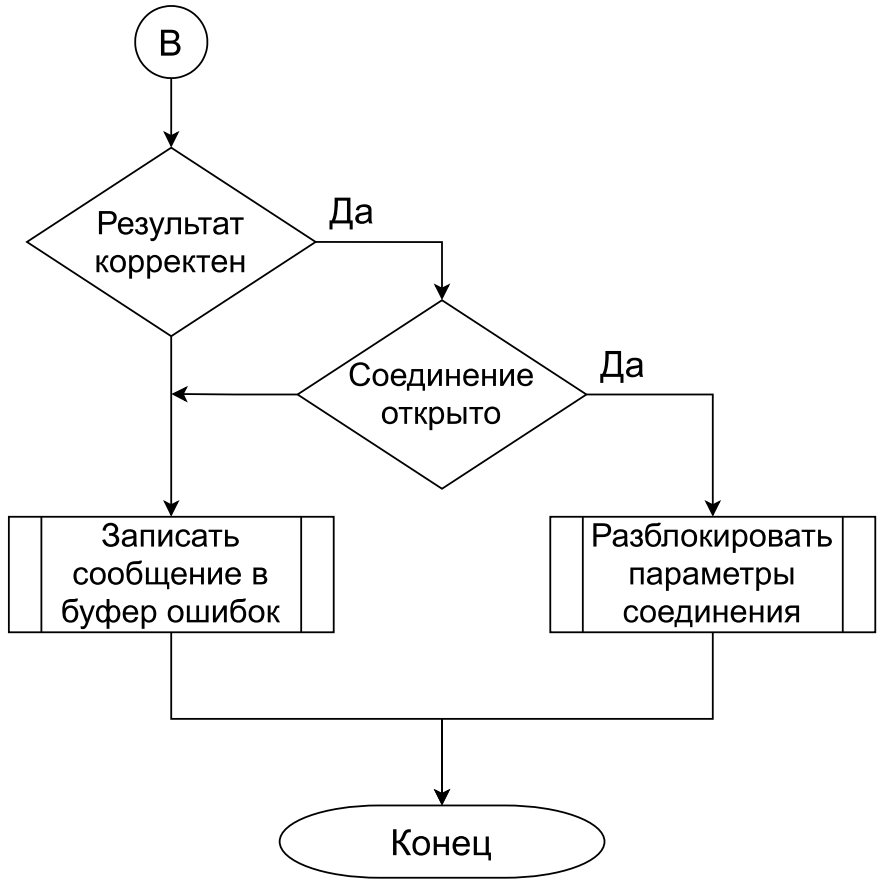
Отправка сообщения:

- размер выходного буфера достиг 8 Кб (исключает отправку маленьких пакетов);
- была вызвана функция *fflush()*.

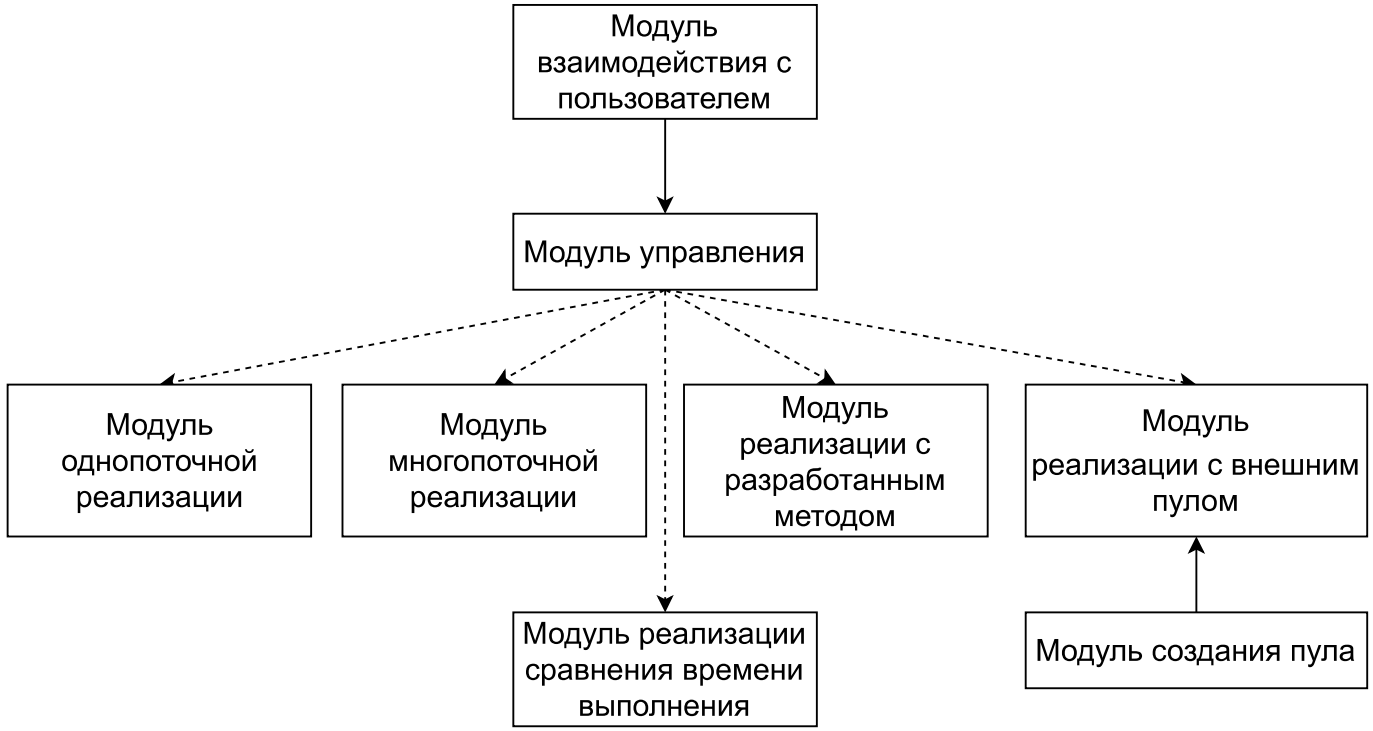
Получение сообщения от сервера







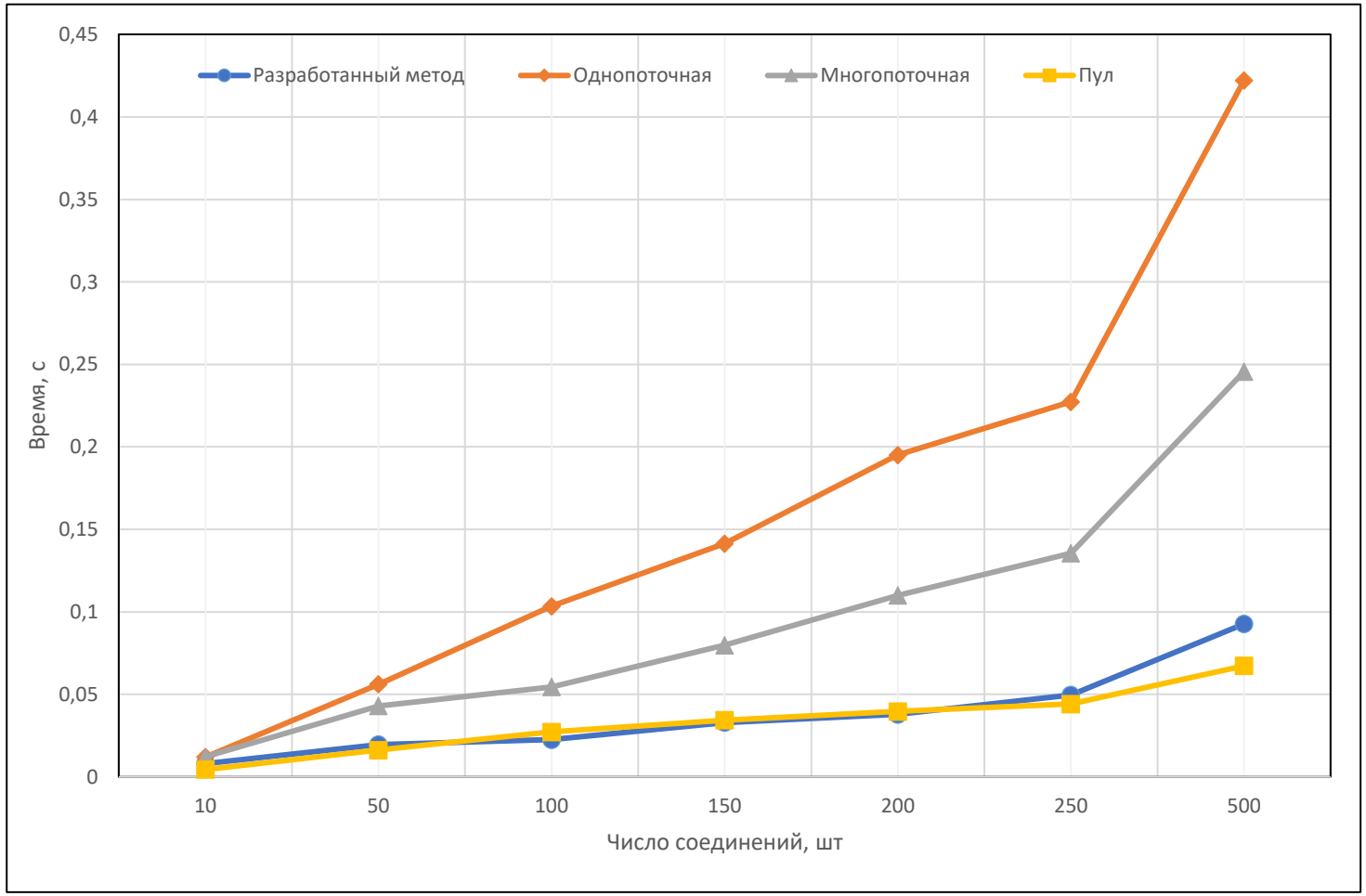
Внешний модуль взаимодействия с разработанным методом



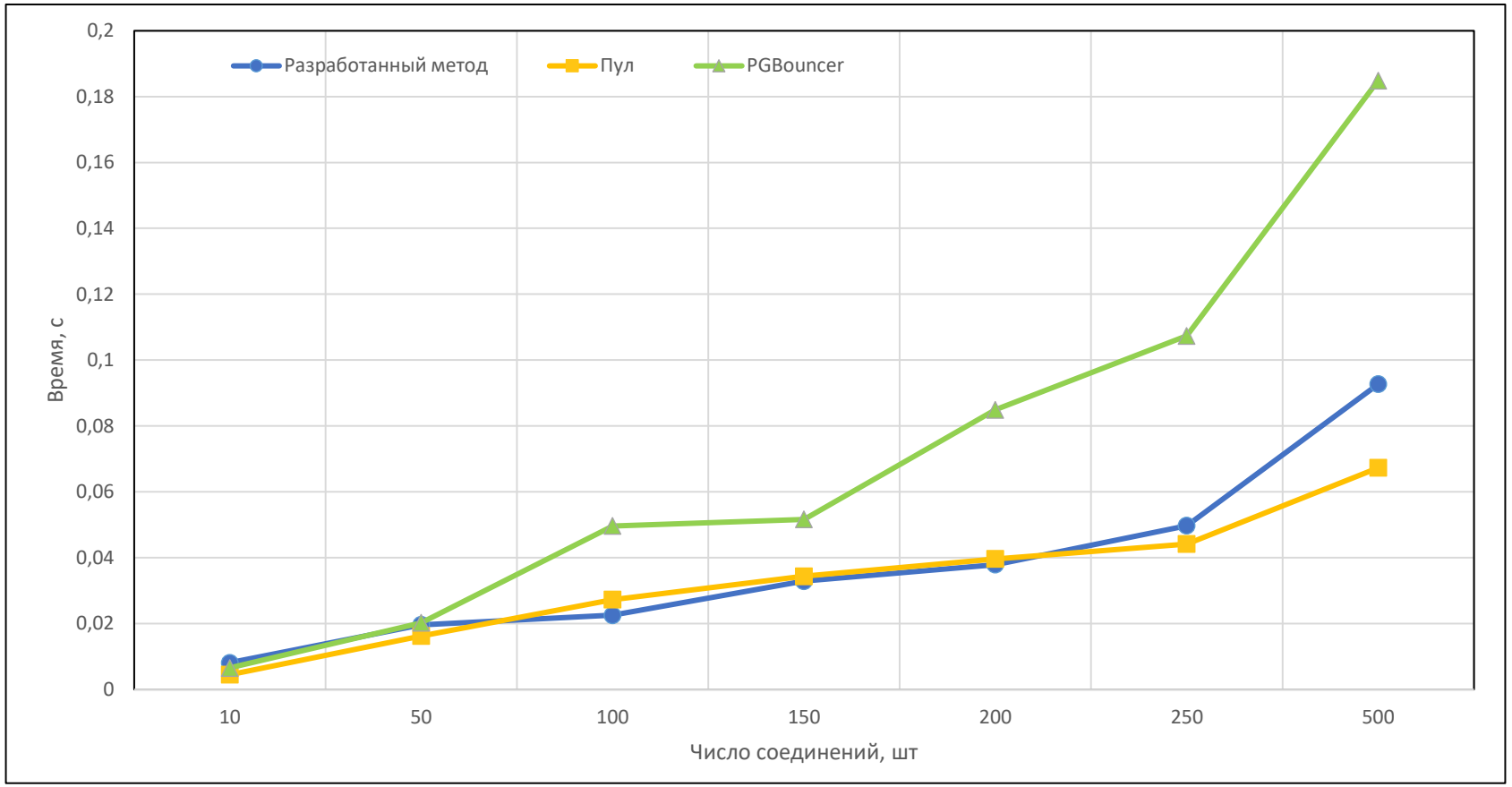
Внешний пул был разработан с использованием умных указателей для предотвращения возможной утечки ресурсов.

Сам пул был реализован в качестве очереди соединений.

Зависимость времени создания соединения от числа подключений к базе данных



Зависимость времени работы реализованного метода и пулов соединений от числа подключений



Исследование требуемых ресурсов для выполнения простого запроса к БД

Простой запрос — выборка всех данных посредством одного оператора ‘SELECT’ и одного оператора ‘FROM’.

SELECT * FROM table100;
Создание 10 соединений.

Реализация	Число раз выделения памяти	Суммарный объем используемой памяти
Однопоточная	729	588,870 байт
Многопоточная	812	593,508 байт
Внешний пул	831	586,212 байт
Разработанный метод	182	180,794 байт

Заключение

Цель достигнута: разработан метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

Поставленные задачи решены:

- Выполнен анализ предметной области и существующих методов выполнения запросов в MPP системах;
- Разработан метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения;
- Реализован разработанный метод;
- Выполнен сравнительный анализ времени работы метода и различных реализаций выполнения запросов.

Дальнейшее развитие

- Реализация пользовательского вывода информации об ошибках в случае конкатенации нескольких запросов в одну команду.
- Разработка метода управления ресурсным пулом в случае потери потоком соединения с базой данных.