



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ
РАБОТЕ
НА ТЕМУ:
«Метод реализации многопоточного доступа к
СУБД»

Студент группы ИУ7-75Б

(Подпись, дата)

О.С. Платонова
(И.О.Фамилия)

Руководитель ВКР

(Подпись, дата)

М.В. Филиппов
(И.О.Фамилия)

Нормоконтролер

(Подпись, дата)

(И.О.Фамилия)

2021 г.

РЕФЕРАТ

Расчетно-пояснительная записка 10 страниц, 3 рисунка, 6 источников.

БАЗА ДАННЫХ, POSTGRESQL, МНОГОПОТОЧНЫЕ СУБД

СОДЕРЖАНИЕ

Введение.....	5
1 Аналитический раздел.....	7
1.1 Выбор СУБД.....	7
1.2 Многопоточность в PostgreSQL.....	7
1.3 Анализ существующих решений.....	8
Список использованных источников	10

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

База данных (БД) – собрание данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения. [1]

Система управления базой данных (СУБД) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. [2]

Введение

В XXI веке человечество владеет невообразимым объемом данных. Знания, передававшиеся из поколения в поколение в течение многих тысячелетий продолжают увеличиваться и по сей день. Так, ежегодный прирост информации составляет 30%. [3]

С появлением письменности, будь то шумерские таблички или берестяные грамоты, перед человечеством возникает вопрос хранения и обработки данных. Причем с развитием цивилизации, и, как следствие, увеличением документооборота, проблема хранения информации требует систематического решения. Например, в конце XX века данные крупной компании могли занимать несколько этажей, что требовало дополнительных кадров для работы с ними.

Первым этапом решения этого вопроса стало внедрение компьютеров. Многие операции с данными были упрощены, а быстрый рост информационных технологий привел к увеличению скорости работы над данными. Однако хранение информации в виде файлов на одном компьютере стало неэффективным. Во-первых, поиск файла в файловой системе был долгим. Во-вторых, хранение информации в одном файле затрудняло поиск необходимых данных.

Решение проблемы разрозненного хранения данных впервые было представлено на симпозиуме в 1963 году в Санта-Монике. Хотя речь шла о внедрении баз данных в военные приложения, этот момент считается точкой отсчета истории базы данных. Их применение в работе компаний привело к увеличению скорости работы. А автоматизация основных процессов базы данных, таких как создание, просмотр, удаление данных привело к созданию системы управления базы данных.

В 2021 году ни одна сфера жизни не обходится без компьютеризации. Организации используют базы и СУБД для перевода данных в электронный вид. Необходимость перевода заключается не столько в потребности сократить временные и материальные (сокращение кадров) расходы, сколько в

поддержании конкурентоспособности. Переход компании в электронный вид дает возможность приобретения принципиально новых качеств, позволяющих иметь существенные преимущества над другими.

Из-за высокой популярности СУБД возникает вопрос об оптимизации ее работы. Так как один из самых распространенных способов увеличения производительности - параллельное выполнение, следует рассмотреть оптимизацию многопоточной программы. Поскольку операция соединения с базой данных является одной из самых долгих, следует минимизировать количество соединений.

Целью данной работы является реализация многопоточного доступа к СУБД. Для достижения поставленной цели необходимо решить следующие задачи:

- исследование и анализ существующих решений;
- исследование возможности распараллеливания подключения;
- реализация распараллеливания;
- анализ и сравнение времени работы для исходного случая и реализуемого.

1 Аналитический раздел

В данном разделе будет выполнен анализ СУБД, представлены существующие методы и алгоритмы решения поставленной задачи. Также будет выполнен анализ решений с указанием достоинств и недостатков.

1.1 Выбор СУБД

На рисунке 1.1 представлен рейтинг популярности СУБД составленный компанией «DB-Engines» по состоянию на конец 2021 года. [4]

381 systems in ranking, November 2021

Rank			DBMS	Database Model	Score		
Nov 2021	Oct 2021	Nov 2020			Nov 2021	Oct 2021	Nov 2020
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1272.73	+2.38	-72.27
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1211.52	-8.25	-30.12
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	954.29	-16.32	-83.35
4.	4.	4.	PostgreSQL + ⓘ	Relational, Multi-model ⓘ	597.27	+10.30	+42.22
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	487.35	-6.21	+33.52
6.	6.	↑ 7.	Redis +	Key-value, Multi-model ⓘ	171.50	+0.15	+16.08
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model ⓘ	167.52	+1.56	+5.90
8.	8.	8.	Elasticsearch	Search engine, Multi-model ⓘ	159.09	+0.84	+7.54
9.	9.	9.	SQLite +	Relational	129.80	+0.43	+6.48
10.	10.	10.	Cassandra +	Wide column	120.88	+1.61	+2.13

Рисунок 1.1 — Рейтинг популярности СУБД.

Согласно рейтингу, лидирующие позиции занимают реляционные модели баз данных. В данной работе будет рассматриваться объектно-реляционная СУБД PostgreSQL 12-ой версии, занимающая 4-ую строчку. Выбор аргументирован доступностью исходного кода, а также кроссплатформенностью системы.

1.2 Многопоточность в PostgreSQL

Каждое соединение представляется объектом PGconn, который можно получить от функций PQconnectdb, PQconnectdbParams или PqsetdbLogin.

PostgreSQL содержит инструменты для реализации многопоточности. Один из них — библиотека libpq, которая по умолчанию поддерживает

повторные вызовы. Однако при реализации многопоточности существует ограничение: «два потока не должны пытаться одновременно работать с одним объектом PGconn. В частности, не допускается параллельное выполнение команд из разных потоков через один объект соединения.». [5]

Таким образом, если в пользовательской программе реализована многопоточность, каждому потоку следует выполнять подключение к БД. Поскольку операция подключения — одна из самых долгих, рост количества потоков может привести к замедлению работы программы: повышенная нагрузка на системные ресурсы и значительное снижение производительности, особенно на многоядерных системах. Это объясняется увеличением конкуренции при обращении множества процессов к ресурсам PostgreSQL.

1.3 Анализ существующих решений

Один из возможных вариантов решения поставленной задачи — пул коннектов. Идея заключается в создании некоторого количества (в зависимости от задачи) соединений, которые будут доступны второстепенным потокам. На рисунках 1.2 — 1.3 представлен цикл соединения с БД без пула и с его использованием соответственно.

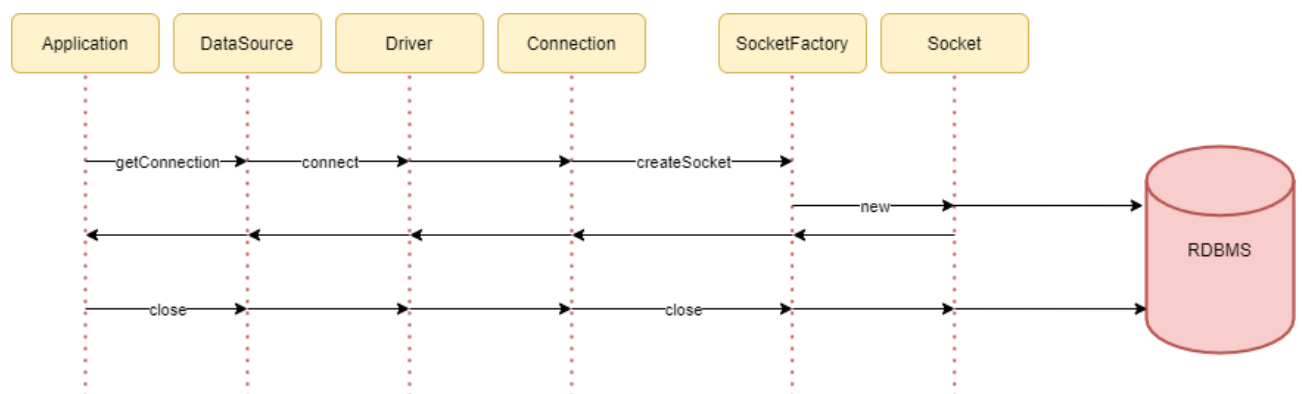


Рисунок 1.2 – Цикл соединения с БД.

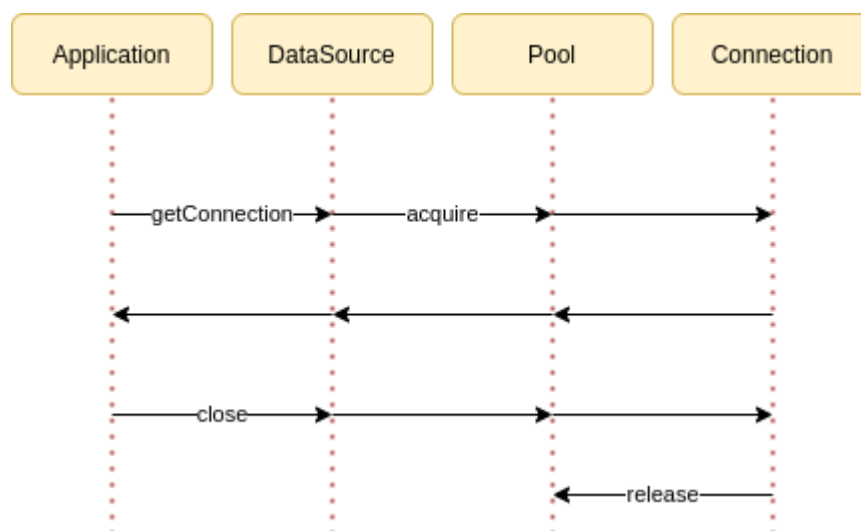


Рисунок 1.3 – Цикл соединения с БД с использованием пула.

Результаты показали, что скорость операций открытия/закрытия соединений с использованием пула может быть увеличена в 600 раз. [6]

К недостаткам данной реализации следует отнести сложность реализации (особенно в крупных корпоративных приложениях), затраты на расчет минимального и максимального размера пула, а также устранение проблем, связанных с его переполнением.

Задача данной работы состоит в реализации случая, при котором главный поток выполняет подключение к БД и передает параметры подключения второстепенным потокам, которые в свою очередь выполняют параллельную работу с БД.

Список использованных источников

1. ГОСТ 34.320-96 Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.
2. ГОСТ Р ИСО/МЭК ТО 10032-2007: Эталонная модель управления данными.
3. Lyman P., Varian H.R. How much information Архивная копия от 19 февраля 2018 на Wayback Machine. Release of the University of California. Oct.27, 2003.
4. Knowledge Base of Relational and NoSQL Database Management Systems. [Электронный ресурс]. - Режим доступа: <https://db-engines.com/en/ranking>.
5. PostgreSQL: Документация: 12.8. [Электронный ресурс]. – Режим доступа: <https://postgrespro.ru/docs/postgresql/12/index>
6. Корсаков А. Б. Анатомия Connection Pooling.