УДК 004.94

Варакута Полина Сергеевна
студентка
Российский экономический университет имени Г.В. Плеханова
Россия, г. Москва

Козлов Роман Константинович
студент
Российский экономический университет имени Г.В. Плеханова
Россия, г. Москва
e-mail: r.kozlov@mail.ru

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОПУСКНОЙ СПОСОБНОСТИ ПУЛОВ СОЕДИНЕНИЙ К БАЗЕ ДАННЫХ POSTGRESQL

Аннотация. *Пулы соединений к базам данных часто используются в высоконагруженных системах, когда необходимо обеспечивать определенный уровень отказоустойчивости и не задействовать лишние финансовые и аппаратные ресурсы. В данной статье проводится анализ модели пула соединений с целью выявления оптимального соотношения затрат и получаемых значений отказоустойчивости и количества процессоров.*

Ключевые слова: компьютерное моделирование, имитационное моделирование, база данных, PostgreSQL, пул соединений, AnyLogic.

Varakuta Polina Sergeevna
student
Russian University of Economics named after G.V. Plekhanov
Russia, Moscow

Kozlov Roman Konstantinovich
student
Russian University of Economics named after G.V. Plekhanov
Russia, Moscow

SIMULATION OF THE CAPACITY OF CONNECTION POOLS TO THE POSTGRESQL DATABASE

Abstract: *Database connection pools are often used in highly loaded systems when it is necessary to provide a certain level of fault tolerance and not to use unnecessary financial and hardware resources. This article analyzes the connection pool model in order to identify the optimal ratio of costs and the resulting values of fault tolerance and the number of processors.*

Key words: computer simulation, simulation modeling, database, PostgreSQL, connection pool, AnyLogic.

При проектировании и построении сложных и высоконагруженных систем немаловажно грамотно подходить к настройке и оптимизации работы с хранилищами данных. Одной из наиболее известных и популярных реляционных баз данных является PostgreSQL – свободная объектно-реляционная СУБД.

Пул соединений в свою очередь является неотъемлемой частью большинства баз данных на PostgreSQL [1]. Он предоставляет механизмы доступа к БД и распределения нагрузки, в результате которых при большой загруженности БД к выполнению запросов допускается только определенная часть клиентских процесс.

Непонимание администраторами приложений принципов работы пула соединений на приложении ведёт к перерасходу технических ресурсов – в частности памяти и процессорного времени. В условиях же ограниченных ресурсов, неверные настройки пула соединений приводят к увеличению как времени отклика приложения, так и количества сбоев, вплоть до полной недоступности сервиса.

Данная работа позволит произвести анализ необходимого минимального размера пулов соединений на серверах приложения и решить ряд связанных с этим проблем:

1. Повышение отказоустойчивости приложения
2. Оптимизация использования ресурсов
3. Масштабирование системы при росте
4. Изменение параметров с учётом добавления/изменения функционала
5. Образовательная часть, наглядная демонстрация принципов работы пулов соединений

Распространено ошибочное мнение, что увеличение размера пула соединений прямо пропорционально влияет на количество успешно выполненных запросов клиентский процессов.

Целями данного исследования являются:

1. опровержение описанного выше мнения,

2. вычисление оптимального размера пулов соединений на серверах приложения.

Имитационное моделирование используется для эффективного принятия решений не только в прикладных предметных областях, таких как различные отрасли народного хозяйства и военное дело, но также и в проектах, связанных с разработкой программного обеспечения и проектированием сложных IT-систем [2, 3]. Из-за чего и было решено использовать имитационное моделирование в AnyLogic в качестве метода исследования.

Наша агентная модель AnyLogic включает в себя три вида агентов: Client, Request и Execution. Для каждого агента существует свой особый путь.

1. Client после создания встает в очередь для получения соединения от пула соединений. При получении соединения создает Request и отправляет его на выполнение базе данных. Ожидает либо возвращения выполненного Request, либо прекращение Request с помощью механизма таймаутов и, освободив соединение, завершает работу.

2. Request после создания переходит на базу данных и создает Execution. Ожидает возвращение выполненного Execution, отрабатывает время транзакции и возвращается к Client.

3. Execution после создания отрабатывает время выполнения запроса и возвращается к Request.

Для настройки модели и проведения экспериментов были введены следующие переменные:

- Кол-во серверов приложения;
- Кол-во пулов соединений;
- Кол-во сессий каждого пула соединений;
- Кол-во CPU базы данных;
- Кол-во клиентских процессов в секунду.

Так как практически все клиентский процессы отличаются друг от друга в модель заложены десять групп клиентских процессов, у которых существуют следующие переменные:

- Таймаут клиентского процесса;
- Номер пула для соединения;
- Доля процессов данной группы от всех процессов;
- Время выполнения транзакции запроса;
- Время работы запросы с базой данных.

В ходе первого эксперимента, направленного на выявление минимального размера пула соединений для 95% отказоустойчивости (доля успешно обработанных клиентских запросов от всех клиентских запросов), были использованы несколько наборов входных параметров. Для каждого набора проявлялось единообразное поведение кривых отказоустойчивости при изменении количества соединений (рис. 1): резкий рост отказоустойчивости при минимальном отклонении количества соединений относительно начала координат. При дальнейшем увеличении количества сессий наблюдается плавный переход графика в плато при приближении к определенному значению (70% при основном наборе данных). Это позволяет нам сделать вывод, что:

1. Бесконечное увеличение соединений на пуле не позволит достичь необходимой 95% отказоустойчивости.

2. Повышение отказоустойчивости зависит не только от количества соединений пула.

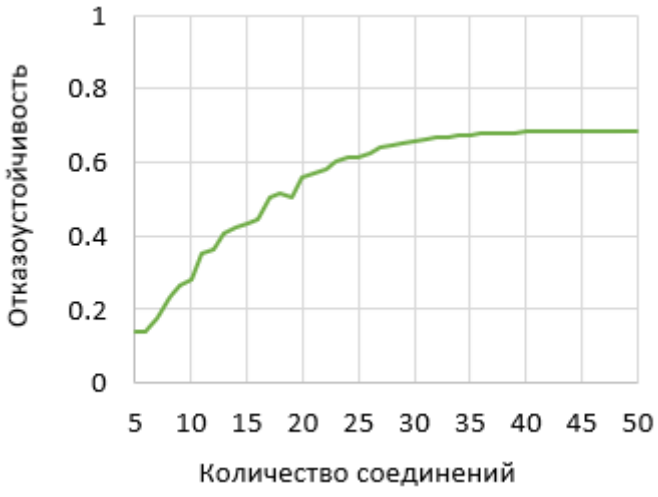

Рисунок 1. Изменение отказоустойчивости при изменении количества соединений на пуле

В ходе анализа полученных результатов первого эксперимента была выдвинута гипотеза, что отказоустойчивость выходит на плато, так как клиентские процессы завершают работу из-за таймаутов. Долгая работа запросов связана с нехваткой CPU и длительным временем выполнения запросов на БД. Исходя из этого, нами был проведен второй эксперимент (рис. 2).

Рисунок 2. Изменение отказоустойчивости при изменении количества CPU и количества соединений на пуле

В ходе второго эксперимента, направленного на выявление зависимости отказоустойчивости от количества CPU и соединений на пуле, были получены следующие результаты: при константном количестве CPU и росте количества соединений отказоустойчивость выходит на плато, как и в первом эксперименте, при увеличении количества CPU и фиксированном количестве соединений отказоустойчивость так же выходит на плато. При одновременном увеличении обоих параметров отказоустойчивость стремится к 100%.

Увеличение количества соединений на пуле не несет финансовых затрат, однако увеличение CPU является не малостоящим событием. Чтобы добиться необходимой отказоустойчивости стоит сперва пересмотреть запросы, выполняющиеся на базе данных, чтобы уменьшить их время выполнения, и только потом задумываться об увеличении количества как соединений на пуле, так и CPU.

Список литературы:

1. PostgreSQL Connection Pool: Part 1 – Pros & Cons. [Электронный ресурс] // Режим доступа: URL: https://scalegrid.io/blog/postgresql-connection-pooling-part-1-pros-and-cons/ (дата обращения: 10.05.2022 г.).

2. de Boer F.S., Grabe I., Jaghoori M.M., Stam A., Yi W., Modeling and Analysis of Thread-Pools in an Industrial Communication Platform // Lecture Notes in Computer Science. 2009. Vol. 5885.

3. Труб И.И. Имитационная модель пула потоков для сервера баз данных. ИММОД. Секционные доклады. 2021. 420 с.