

УДК 004

**РАСШИРЕНИЕ ИНДЕКСА ТИПА В-ДЕРЕВО С ВЫДЕЛЕНИЕМ  
МНОЖЕСТВА НОВЫХ СТРАНИЦ В POSTGRESQL**

Лицкевич К.Д., студент гр. 628, VI курс

Научный руководитель: Хорошилов А.В., к.ф.-м.н., доцент  
Московский государственный университет имени М.В. Ломоносова  
г. Москва

В современном мире, где объемы данных растут с каждым днем, эффективное хранение и оптимальная работа с информацией становятся критически важными для успешной деятельности организаций и предприятий, а системы управления базами данных (СУБД) являются неотъемлемой частью IT-инфраструктуры. СУБД определяется как система управления данными, позволяющая работать с несколькими файлами, обеспечивая их согласованность [1]. К производительности систем управления базами данных предъявляются высокие требования, и оптимизация структуры хранения и поиска данных становится одной из ключевых задач для разработчиков и исследователей.

Одним из способов повышения производительности поиска и доступа к данным является использование индексов. В обычном случае поиск в таблице записей, соответствующих заданным критериям, производится путём последовательного просмотра всей таблицы. В случае, если таблица имеет большое количество строк, такой поиск может занимать много времени. Индексы часто сравнивают с алфавитным указателем в книге или каталогом в библиотеке, он позволяет быстро находить нужную информацию без необходимости просматривать все данные таблицы. Если про большую таблицу известно, что в значительной части запросов к ней в качестве критерия выступают ограничения на значения в определённых полях, для ускорения выполнения таких запросов можно построить индекс для этой таблицы по этим полям. Индекс формируется из значений одного или нескольких полей таблицы и указателей на соответствующие строки таблицы и имеет структуру, оптимизированную под поиск — например, сбалансированное дерево. Эффективное использование индексов позволяет значительно улучшить производительность запросов, особенно для таблиц с большим объемом данных. Индексы также могут использоваться для обеспечения уникальности значений в столбцах таблиц, что является критически важным для поддержания целостности данных.

Наиболее распространенной структурой для реализации индексов являются В-дерева [2] и их модификации. Эти структуры обеспечивают логически упорядоченное хранение данных, позволяя выполнять операции поиска, вставки и удаления за логарифмическое время.

PostgreSQL [3, 4, 5] – одна из ведущих объектно-реляционных СУБД. Это свободно распространяемый продукт с открытым исходным кодом, который доступен на большом числе платформ. PostgreSQL поддерживает почти все основные возможности стандарта SQL:11 и предоставляет практически полезные дополнительные возможности. PostgreSQL поддерживает несколько типов индексов, в том числе и В-деревья [6]. При создании индекса в PostgreSQL по умолчанию используется именно тип В-дерево, поскольку он эффективен в большинстве случаев. Кроме того В-дерево – это единственный тип индекса в PostgreSQL, поддерживающий обеспечение уникальности значений. Когда для таблицы определяется ограничение уникальности или первичный ключ, PostgreSQL автоматически создаёт уникальный индекс типа В-дерево по всем столбцам, составляющим это ограничение или первичный ключ (индекс может быть составным). Таким образом В-деревья активно используются в PostgreSQL.

PostgreSQL хранит таблицы и индексы в файлах на диске, при этом каждый файл логически поделен на страницы (также называемые блоками) для удобства организации ввода-вывода. До 2016 года, когда при вставке очередной записи в таблицу заканчивалось место на уже выделенных для этой таблицы страницах и требовалось увеличить файл, он увеличивался на одну страницу за раз. Увеличивать файл таблицы необходимо, удерживая блокировку на расширение таблицы. Эта блокировка становится узким местом, когда в таблицу происходит вставка сразу большого количества записей в параллельных потоках.

В 2016 году, начиная с PostgreSQL версии 9.6 для улучшения масштабируемости СУБД была оптимизирована параллельная вставка. Идея оптимизации заключалась в том, что в случае, когда несколько процессов ожидали блокировку на расширение таблицы, процесс, взявший блокировку перепроверял наличие свободных страниц и в случае их отсутствия увеличивал файл таблицы сразу на несколько страниц, пропорционально количеству процессов, ожидающих освобождения блокировки. Большое число ожидающих блокировку процессов интерпретируется при этом как то, что в данную таблицу активно производится запись, и что вскоре другим процессам также понадобится место для вставки новых записей, которое они найдут на выделенных наперёд страницах, так что расширять таблицу им не требуется, как и захватывать блокировку.

Несмотря на то, что один процесс выделял сразу несколько страниц, он всё ещё выделял их по одной, для каждой совершая отдельный системный вызов. В 2023 году в PostgreSQL 16 это было исправлено, появилась возможность увеличивать файл таблицы сразу на указанное число страниц с помощью одного системного вызова. Кроме того изменился способ расчёта количества страниц, которые нужно выделять: у процесса, производящего вставку, появилась возможность запрашивать сразу несколько страниц, которые он гарантированно получит после выполнения увеличения файла таблицы, и это число запрошенных страниц умножается на количество

ожидающих блокировку процессов, чтобы определить количество страниц, которые необходимо выделить дополнительно для других процессов.

Оптимизация выделения новых страниц для таблицы в PostgreSQL позволила значительно ускорить вставку большого числа записей параллельно. Тем не менее, увеличение файла для индекса по-прежнему осуществляется по одной странице, что приводит к снижению производительности на высоконагруженных системах в условиях параллельной вставки большого числа данных множеством процессов в таблицу с индексом. Так же, как это раньше происходило с блокировкой на расширение таблицы, теперь процессы скапливаются в ожидании блокировки на расширение индекса. Если в данный момент лишь исключительные пользователи СУБД работают с достаточно большим объёмом данных, чтобы наблюдать данную проблему, то в будущем, с увеличением объёмов данных, генерируемых информационными системами, число пользователей СУБД, для которых решение этой проблемы станет критически важным, будет неумолимо расти. В PostgreSQL индекс типа В-дерево, как наиболее используемый, требует оптимизации выделения новых страниц в первую очередь.

Требуется исследовать реализованный в PostgreSQL механизм расширения таблицы с выделением множества новых страниц и реализовать в PostgreSQL механизм расширения индекса типа В-дерево с выделением множества новых страниц.

В PostgreSQL реализован буферный кэш для сглаживания разницы между скоростью доступа к данным на диске и в оперативной памяти. Он располагается в общей памяти сервера, доступен всем процессам и представляет собой массив буферов, в каждом из которых содержатся данные одной страницы. Любой доступ к странице осуществляется через менеджер буферов. Менеджер буферов производит считывание новых страниц с диска в буферный кэш, вытеснение из кэша страниц, к которым давно не было обращений, запись грязных страниц на диск и прочие операции буферного кэша.

Для корректной работы параллельных процессов сервера со страницами в буферном кеше и корректной записи этих страниц на диск существует система правил о взятии блокировок и выставлении флагов на страницах. Согласно этим правилам должны работать как менеджер буферов, так и модули, обращающиеся к нему.

Механизм расширения таблицы с выделением множества новых страниц состоит из двух частей:

1. функционал в составе менеджера буферов, производящий выделение новых страниц и их закрепление в буферном кэше;
2. функционал в составе модуля ввода-вывода таблиц, обращающийся к менеджеру буферов и выполняющий все остальные действия, необходимые для будущего использования страниц, с учётом устройства страниц таблиц и правил взаимодействия с ними.

Менеджер буферов – это один из базовых компонентов PostgreSQL, функции которого вызываются из различных других частей СУБД. Поэтому при разработке механизма расширения индексов типа В-дерево с выделением множества новых страниц было принято решение использовать функционал менеджера буферов, по возможности избегая внесения изменений в него. С учётом этого необходимо реализовать функционал в составе модуля управления страницами В-дерева, аналогичный упомянутому выше функционалу в составе модуля ввода-вывода таблиц, который будет обращаться к менеджеру буферов для выделения новых страниц и подготавливать эти страницы для будущего использования, с учётом особенностей В-дерева.

Одним из основных различий между таблицей и индексом заключается в том, что в таблице не гарантируется порядок, в котором хранятся записи, а индекс обязан поддерживать заданный порядок записей и его страницы образуют определённую поисковую структуру. В таблицах при массовом добавлении данных процессу заранее известны записи, которые необходимо добавить в таблицу, и возможно спрогнозировать число страниц, которое понадобится для осуществления вставки всех данных. В случае же индекса типа В-дерево, вставка каждой новой записи происходит независимо от остальных, и для каждой записи находится конкретная страница, на которую она должна быть вставлена. В случае, если на этой странице недостаточно свободного пространства, происходит разделение (англ. split) этой страницы, при котором выделяется новая страница. Из-за этого в случае индекса сложнее спрогнозировать, сколько страниц может понадобиться для вставки всех данных. Кроме того, между двумя вставками, требующими выделения новой страницы, может пройти достаточно долгое время. По этим причинам преимущества от выделения сразу нескольких страниц для одного процесса во время расширения индекса типа В-дерево остаются неясными. В данной работе вопрос возможности выделения нескольких новых страниц для одного процесса при работе с индексом типа В-дерево не рассматривается, вместо этого акцент делается на возможность выделения дополнительных страниц для других процессов.

Кроме того, у страниц индекса типа В-дерево и страниц таблицы существенно отличаются логика начальной инициализации и блокировки из-за чего невозможно переиспользовать функционал из модуля ввода-вывода таблиц, а необходимо реализовывать отдельный функционал в модуле управления страницами В-дерева.

Реализованный механизм расширения индекса типа В-дерево с выделением множества новых страниц позволил повысить производительность PostgreSQL в условиях параллельной вставки большого числа данных множеством процессов в таблицу с индексом типа В-дерево.

Полученную реализацию механизма расширения таблицы с выделением множества новых страниц планируется предложить в сообщество PostgreSQL для включения в кодовую базу СУБД.

**Список литературы:**

1. Кузнецов С. Д. Основы баз данных / С. Д. Кузнецов. — Москва : Национальный Открытый Университет ИНТУИТ, 2016. — 350 с.
2. Bayer R., McCreight E. Organization and maintenance of large ordered indices // Proceedings of the 1970 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control. — 1970. — С. 107–141.
3. Stonebraker M., Rowe L. A. The design of POSTGRES // Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data. — 1986. — С. 340–355.
4. Рогов Е. В. PostgreSQL 16 изнутри. — М.: ДМК Пресс, 2024. — 664 с.
5. PostgreSQL 16.4 Documentation / The PostgreSQL Global Development Group, 1996–2024 [Электронный ресурс] // postgresql.org URL: <https://www.postgresql.org/docs/16/index.html> (дата обращения: 28.03.2025).
6. Рогов Е. Индексы в PostgreSQL — 4 [Электронный ресурс] // habr.com URL: <https://habr.com/ru/companies/postgrespro/articles/330544/> (дата обращения: 28.03.2025).