

Методическое пособие по теме:

«Проверка скорости работы запросов и их оптимизация с помощью внешней составной искусственной индексации»

Цель: Проверка скорости работы запросов и их оптимизация с помощью внешней составной искусственной индексации

Содержание:

1. Pgbench.....1
2. Создание индексов.....2

Pgbench:

Команда `pgbench` — запустить тест производительности PostgreSQL

Пример

```
pgbench -i [параметр...] [имя_бд]
pgbench [параметр...] [имя_бд]
```

Команда `pgbench` — это простая программа для запуска тестов производительности PostgreSQL. Она многократно выполняет одну последовательность команд, возможно в параллельных сеансах базы данных, а затем вычисляет среднюю скорость транзакций (число транзакций в секунду). По умолчанию `pgbench` тестирует сценарий, примерно соответствующий TPC-B, который состоит из пяти команд `SELECT`, `UPDATE` и `INSERT` в одной транзакции.

Для запускаемого по умолчанию теста типа TPC-B требуется предварительно подготовить определённые таблицы. Чтобы создать и наполнить эти таблицы, следует запустить `pgbench` с ключом `-i` (инициализировать).

Запуск инициализации выглядит так:

```
pgbench -i [ другие-параметры ] имя_базы
```

где `имя_базы` — имя уже существующей базы, в которой будет проводиться тест. (Чтобы указать, как подключиться к серверу баз данных, вы также можете добавить параметры `-h`, `-p` и/или `-U`.)

pgbench -i

Создаёт четыре таблицы: `pgbench_accounts`, `pgbench_branches`, `pgbench_history` и `pgbench_tellers`, предварительно уничтожая существующие таблицы с этими именами.

С «коэффициентом масштаба», по умолчанию равным 1, эти таблицы изначально содержат такое количество строк:

table	# of rows

<code>pgbench_branches</code>	1
<code>pgbench_tellers</code>	10
<code>pgbench_accounts</code>	100000
<code>pgbench_history</code>	0

Эти числа можно (и в большинстве случаев даже нужно) увеличить, воспользовавшись параметром `-s`(коэффициент масштаба). При этом также может быть полезен ключ `-F` (фактор заполнения).

Подготовив требуемую конфигурацию, можно запустить тест производительности командой без `-i`, то есть:

```
pgbench [ параметры ] имя_базы
```

Практически во всех случаях, чтобы получить полезные результаты, необходимо передать какие-либо дополнительные параметры. Наиболее важные параметры: `-c` (число клиентов), `-t` (число транзакций), `-T`(длительность) и `-f` (файл со скриптом). `pgbench` принимает следующие аргументы командной строки для **тестирования производительности**:

`-c` клиенты

`--client=клиенты`

Число имитируемых клиентов, то есть число одновременных сеансов базы данных. Значение по умолчанию — 1.

`-C`

`--connect`

Устанавливать новое подключение для каждой транзакции вместо одного для каждого клиента. Это полезно для оценивания издержек подключений.

`-d`

`--debug`

Выводить отладочные сообщения.

-j потоки

--jobs=потоки

Число рабочих потоков в `rgbench`. Использовать нескольких потоков может быть полезно в многоядерных системах. Число клиентов должно быть кратно числу потоков, так как каждый поток должен управлять одинаковым числом клиентских сеансов. Значение по умолчанию — 1.

-l

--log

Записать время выполнения каждой транзакции в файл протокола.

CREATE INDEX

Индекс (англ. *index*) — объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет искать строки, удовлетворяющие критерию поиска. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск — например, сбалансированного дерева.

CREATE INDEX — создать индекс

```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ [ IF NOT EXISTS ] имя ] ON имя_таб-
лицы [ USING метод ]
    ( { имя_столбца | ( выражение ) } [ COLLATE правило_сортировки ] [ класс_
операторов ] [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [, ...] )
    [ INCLUDING ( имя_столбца [, ...] ) ]
    [ WITH ( параметр_хранения = значение [, ...] ) ]
    [ TABLESPACE табл_пространство ]
    [ WHERE предикат ]
```

CREATE INDEX создаёт индексы по указанному столбцу(ам) заданного отношения, которым может быть таблица или материализованное представление. Индексы применяются в первую очередь для оптимизации производительности базы данных (хотя при неправильном использовании возможен и противоположный эффект).

Выражение в предложении **WHERE** может ссылаться только на столбцы нижележащей таблицы, но не обязательно ограничиваться теми, по которым

строится индекс. В настоящее время в WHERE также нельзя использовать подзапросы и агрегатные выражения. Это же ограничение распространяется и на выражения в полях индексов.

Параметры

UNIQUE

Указывает, что система должна контролировать повторяющиеся значения в таблице при создании индекса (если в таблице уже есть данные) и при каждом добавлении данных. Попытки вставить или изменить данные, при которых будет нарушена уникальность индекса, будут завершаться ошибкой.

CONCURRENTLY

С этим указанием Postgres Pro построит индекс, не устанавливая никаких блокировок, которые бы предотвращали добавление, изменение или удаление записей в таблице; тогда как по умолчанию операция построения индекса блокирует запись (но не чтение) в таблице до своего завершения.

IF NOT EXISTS

Не считать ошибкой, если индекс с таким именем уже существует. В этом случае будет выдано замечание. Заметьте, что нет никакой гарантии, что существующий индекс как-то соотносится с тем, который мог бы быть создан. Имя индекса является обязательным, когда указывается IF NOT EXISTS.

NCLUDING

Необязательное предложение INCLUDING позволяет указать список столбцов, которые будут включены в неключевую часть индекса. Столбцы, указанные в этой части, не могут одновременно входить в ключевую часть, и наоборот. Выражение INCLUDING предназначено только для того, чтобы больше запросов могли быть ускорены путём использования сканирования только индекса.

имя

Имя создаваемого индекса.

имя_таблицы

Имя индексируемой таблицы

метод

Имя применяемого метода индекса. Возможные варианты: btree, hash, gist, spgist, gin и brin. По умолчанию подразумевается метод btree.

имя_столбца

Имя столбца таблицы.

выражение

Выражение с одним или несколькими столбцами таблицы.

правило_сортировки

Имя правила сортировки, применяемого для индекса. По умолчанию используется правило сортировки, заданное для индексируемого столбца, либо полученное для результата выражения индекса. Индексы с нестандартными правилами сортировки могут быть полезны для запросов, включающих выражения с такими правилами.

класс_операторов

Имя класса операторов.

ASC

Указывает порядок сортировки по возрастанию

DESC

Указывает порядок сортировки по убыванию.

NULLS FIRST

Указывает, что значения NULL после сортировки оказываются перед остальными. Это поведение по умолчанию с порядком сортировки DESC.

NULLS LAST

Указывает, что значения NULL после сортировки оказываются после остальных. Это поведение по умолчанию с порядком сортировки ASC.

параметр_хранения

Имя специфичного для индекса параметра хранения

табл_пространство

Табличное пространство, в котором будет создан индекс. Если не определено, выбирается default_tablespace, либо temp_tablespaces, при создании индекса временной таблицы.

предикат

Выражение ограничения для частичного индекса.

CREATE INDEX является языковым расширением Postgres Pro. Средства обеспечения индексов в стандарте SQL не описаны.

Вывод: В данной лабораторной работе рассмотрены индексация и проверка скорости работы запросов в базе данных.

Контрольные вопросы:

1. Что такое pgbench ?
2. Что такое индекс?

Задания для выполнения:

1. Запустить тест производительности для базы данных без индексов.
2. Добавить в таблицы базы данных 10000 записей
3. Запустить тест производительности для базы данных с большим количеством записей в таблицах.
4. Проиндексировать таблицы и запустить тест производительности.