

ІНДЭКСАЦЫЯ БАЗ ДАДЗЕННЫХ

Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі
г. Мінск, Рэспубліка Беларусь

Дзікі А.А.

Стройнікава А. Д. – асістэнт кафедры інфарматыкі

Рана ці позна прыходзіць момант, калі прадуктыўнасць базы дадзеных больш не здавальняючая. Індэкс – гэта структура дадзеных, якая дазваляе аптымізаваць пошук і доступ да дадзеных. Калі пачынае расці база дадзеных, яе прадуктыўнасць будзе праблемай. Такім чынам, прыярытэтай задачай выкарыстання баз дадзеных з'яўляецца атрыманне непасрэдна канкрэтнага радку ў вялікай табліцы ў мінімальна магчымыя тэрміны.

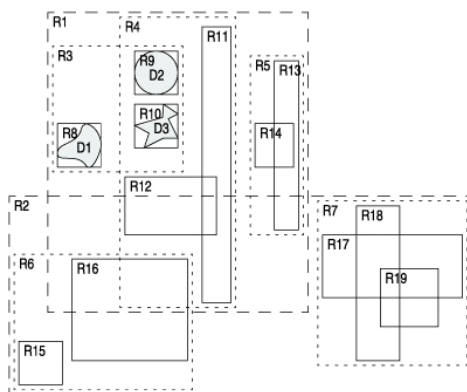
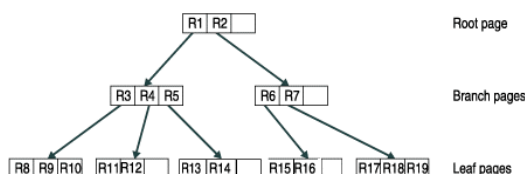
Індэксацыя ўяўляе сабой метадавую структуру дадзеных для эфектыўнага здабывання запісаў з файлаў базы дадзеных на аснове некаторых атрыбутаў, на якіх была зроблена індэксацыя. Індэкс можа быць кластарызаваны (радкі дадзеных сартуюцца па ключавым полі) і некластарызаваны (дадзеныя захоўваюцца ў кучы).

Найбольш папулярнай структурай дадзеных для аднамернага (іерархічнага) індэксу з'яўляецца В+-дрэва. Яго можна разглядаць як В-дрэва. Аднак замест таго, каб разглядаць усе вузлы роўнымі, дадзеная структура мае два тыпу вузлоў: ліставыя вузлы (утрымваюць фактычныя дадзеныя) і астатнія, уключаючы каранёвы вузел, якія ўтрымваюць толькі ключавыя значэнні і паказальнікі на наступныя вузлы. Дадаваны тып аптымальны для мноства з добрым размеркаваннем значэнняў і высокай магутнасцю.

Для арганізацыі хуткага пошуку ў прасторавых дадзеных, напрыклад геаграфічныя дадзеныя з двухмернымі каардынатамі (шырынёй і даўжынёй), выкарыстоўваюцца індэксы прасторавай сеткі (Spatialgrid) і R*-дрэвы.

Індэкс прасторавай сеткі дзеліць вобласць на лагічныя квадратныя сеткі з фіксаваным памерам, якія задаюцца пры стварэнні індэкса. Можна вызначыць да трох роўняў прасторавага індэкса (роўні сеткі). Выкарыстанне некалькіх роўняў сеткі карыснае, бо дазваляе аптымізаваць індэкс для розных памераў прасторавых дадзеных. Прастаравае індэкс на аснове сеткі мае тую перавагу, што структура індэкса можа быць створана першай, а дадзеныя дадаюцца на сталай падставе, не патрабуючы якіх-небудзь змен у структуры індэкса.

R*-дрэва (мал. 1) таксама выкарыстоўваецца для геапрасторавых дадзеных. Яно арганізавана падобна да Spatial grid, але, ў адрозненне ад Spatial Grid, не абавязана цалкам пакрываць бацькоўскія ячэйкі, і яны могуць перасякацца. Індэкс выкарыстоўвае абмежавальны прамавугольнік, які з'яўляецца прамалінейнай формай, якая цалкам утрымвае абмежаваны аб'ект ці аб'екты. Абмежавальныя палі могуць складаць аб'екты дадзеных ці іншыя абмежавальныя палі. Абмежавальныя палі звычайна захоўваюцца як набор каардынат аднолькавай памернасці, як абмежаваны аб'ект. Хоць з меркаваннем прадукцыйнасці карысна выбраць абмежавальны прамавугольнік, які як мага менш. Напрыклад, мінімальны абмежавальны прамавугольнік для двухмернай акружнасці – гэта квадрат, бок якога роўны дыяметру круга. Мінімальным абмежавальным прамавугольнікам для трохмернай сферы з'яўляецца куб, край якога роўны дыяметру сферы.



Мал. 1 – Структура індэкса R-дрэва

Перавагай R-дрэва з'яўляецца тое, што яно лягчэй ў рэалізацыі, патрабуе менш дыскавай прасторы, можа выкарыстоўвацца для шматмерных прастораў (Spatialgrid толькі для двухмерных) і пры ўзросце памеру

адсутнічае пагаршэнне прадукцыйнасці. Яго нястачай з'яўляецца праблема апраксімацыі палігонаў і неабходнасць стварэння індэкса для ўсёй Зямлі.

Для параўнання або пабудовы індэксаў для радковых і двайковых дадзеных можна выкарыстоўваць хэш-індэкс. Яны прадгледжваюць захоўванне не саміх значэнняў, а іх хэшаў, дзякуючы чаму змяняецца памер (а, адпаведна, і павялічваецца хуткасць апрацоўкі) індэксаў з вялікіх палёў. Такі тып індэксаў карысны пры аперацыях сартавання і параўнання дадзеных, аднак, выкарыстоўваючы яго, нельга вызначыць, колькі прыкладна радкоў ёсць паміж двума значэннямі, і для яго выкарыстання патрабуюцца толькі цэлыя значэнні ключоў.

Там, дзе ёсць вялікія мноства з нізкай магутнасцю і добрай кластарызацыяй па іх значэннях, выкарыстоўваюцца бітавыя індэксy (bitmap). Бітавыя індэксy выкарыстоўваюць бітавыя масівы і вынікі выконвання пабітавых лагічных аперацый на гэтых масівах. Гэты індэкс эфектыўны пры вялікіх табліцах і калі слупкі часта ўдзельнічаюць у аперацыях ўстаўкі / абнаўлення / выдалення. Аднак, пры аднаўленні мноства бітаў спатрэбіцца час, каб выканаць DML-аперацыю і аднавіць індэксy.

Для манатонна нарастальных значэнняў (напрыклад аўтаінкараментны ідэнтыфікатар) выкарыстоўваюць індэкс зваротнага ключа. Індэкс зваротнага ключа, па параўнанні са стандартным індэксам, звяртае кожны байт слупка, захоўваючы пры гэтым парадак слупкоў. Калі слупок індэксуецца ў зваротным рэжыме, значэнні слупка будуць захоўвацца ў індэксе ў розных блоках па меры таго, як зыходнае значэнне адрозніваецца. Такая кампануюка можа дапамагчы пазбегнуць пагаршэння прадукцыйнасці ў індэксах, дзе змены індэкса сканцэнтраваны на невялікіх наборах блокаў. Індэксy зваротнага ключа змяняюць «гарачыя пункты» ў індэксах, асабліва індэксy першасных ключоў, шляхам змены байтаў блокаў ліста. І, такім чынам, знішчаюць канкурэнцыю за ліставыя блокі па ўсіх асобніках.

Пры выбары тыпу індэкса трэба ўлічваць структуру дадзеных, якія індэксуюцца, а таксама характар працы з імі. Кожны тып па-свойму унікальны, моцны і карысны. І няма ўніверсальнага алгарытму і структуры дадзеных, якія былі б карысны ўсім. Важна памятаць, што выбар тыпу індэксавання можа як дапамагчы вырашыць праблему, гэтак і нашкодзіць, дадаўшы дадатковы расход памяці і замаруджванне часу апрацоўкі запытаў.

Спіс выкарыстаных крыніц:

1. Рассел, Д. Индекс (базы данных) / Д. Рассел, Р. Кон. – М. : Оникс, 2013. – 520 с.
2. Цэнтр ведаў IBM [Электронны рэсурс]. – 2018. – Рэжым доступу: <https://www.ibm.com/support/knowledgecenter>.

СИЛЬНЫЕ СТОРОНЫ ФРЕЙМВОРКА ANGULAR

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Харитонов Н.В., Дроздов А.С.

Стержанов М.В. – к.т.н., доцент

В современном мире, где разработка программного обеспечения развивается быстрыми темпами, наибольшую популярность приобретают веб-приложения. Основными плюсами использования веб-приложений являются простота, удобство, доступность и скорость. Для реализации таких приложений существует множество различных фреймворков разработки, среди которых одним из наиболее популярных в последние три года является Angular.

Angular – Javascript-фреймворк, созданный на основе TypeScript, который позволяет разрабатывать сложные SPA-приложения (Single Page Application). Написание приложения состоит из создания шаблонов с помощью языка разметки HTML, написания классов-компонентов для работы с данными шаблонами, а также выделения общей логики приложения в сервисы. Все вместе это можно объединять в отдельные модули. Angular 2 написан в соответствии с шаблоном проектирования «модуль», что позволяет изолировать части логики приложения от глобального контекста, и содержит собственную систему модулей (NgModules) с тем, чтобы приложение имело понятную структуру и позволяло переиспользовать необходимые части без циклических зависимостей. Разработанный и поддерживаемый компанией Google, он описывается как JavaScript MVW-фреймворк. Angular (он же – Angular 2+, он же – Angular 2 или ng2) является переписанным преемником AngularJS (он же – Angular.js или AngularJS 1.x). Несмотря на то, что AngularJS (ранняя версия) был выпущен в октябре 2010 года, его создатели до сих пор устраняют недоработки данного фреймворка. Новый же Angular (без окончания «JS») был выпущен в свет в сентябре 2016 года в качестве версии №2 своего предка. На данный момент самой последней версией является 5-я. Фреймворк Angular используется такими компаниями, как Google, Wix, weather.com, healthcare.gov и Forbes.

Традиционно, веб-приложение на Angular 2 состоит из набора компонентов (виджетов), которые образуют древовидную структуру. В ней, в свою очередь, родительский компонент имеет ссылку на все вложенные в него дочерние. Обмен данными реализован следующим способом. При получении данных родитель отправляет их часть дочерним компонентам. Они, в свою очередь, либо передают их часть дальше по дереву компонентов, либо отображают эти данные с использованием различных элементов интерфейса. Также, в Angular 2 дочерние компоненты имеют возможность оповестить родительский о различных пользовательских событиях: клик мыши или нажатие клавиши. Также возможно внедрение так называемых сервисов (services) - специальных объектов, предоставляющих компоненту возможность получить данные в