ДОДАТОК А

Лістинг фрагментів коду програми

```
namespace Perst.Impl
{
 using System;
 using System.Collections;
 using System.Diagnostics;
 using Perst;

 enum Btreeresult {
 Done,
 Overflow,
 Underflow,
 Notfound,
 Duplicate,
 Overwrite
 }

 [Serializable]
 internal class Btree:Persistentcollection, Index
 {
 internal int root;
 internal int height;
 internal Classdescriptor.Fieldtype type;
 internal int nelems;
 internal bool unique;
 [Nonserialized()]
 internal int updatecounter;

 internal static int Sizeof = Objectheader.Sizeof + 4 * 4 + 1;

 internal Btree()
 {
 }

 internal Btree(byte[] obj, int offs)
 {
 height = Bytes.unpack4(obj, offs);
 offs += 4;
 nelems = Bytes.unpack4(obj, offs);
 offs += 4;
 root = Bytes.unpack4(obj, offs);
 offs += 4;
 type = (Classdescriptor.Fieldtype)Bytes.unpack4(obj, offs);
 offs += 4;
 unique = obj[offs] != 0;
 }

 internal Btree(Type cls, bool unique)
 {
 type = checktype(cls);
 this.unique = unique;
 }
```

```
        internal Btree(Classdescriptor.Fieldtype type, bool unique)
        {
        this.type = type;
        this.unique = unique;
        }

        public virtual void init(Type cls, Classdescriptor.Fieldtype[]
types, string[] fieldnames, bool unique, long autoinccount)
        {
        this.type = types[0];
        this.unique = unique;
        }

        static protected Classdescriptor.Fieldtype checktype(Type c)
        {
                Classdescriptor.Fieldtype          elemtype          =
Classdescriptor.converttonotnullable(Classdescriptor.gettypecode(c))
;
        if ((int)elemtype > (int)Classdescriptor.Fieldtype.tpoid
        && elemtype != Classdescriptor.Fieldtype.tparrayofbyte
        && elemtype != Classdescriptor.Fieldtype.tpdecimal
        && elemtype != Classdescriptor.Fieldtype.tpguid)
        {
                                    throw                                  new
Storageerror(Storageerror.Errorcode.UNSUPPORTED_INDEX_TYPE, c);
        }
        return elemtype;
        }

        public virtual int comparebytearrays(byte[] key, byte[] item,
int offs, int length)
        {
        int n = key.Length >= length ? length : key.Length;
        for (int i = 0; i < n; i++)
        {
        int diff = key[i] - item[i+offs];
        if (diff != 0)
        {
        return diff;
        }
        }
        return key.Length - length;
        }

        public override int Count
        {
        get
        {
        return nelems;
        }
        }
```

```
public bool Isunique
{
get
{
return unique;
}
}

public int Headersize
{
get
{
return Sizeof;
}
}


public Classdescriptor.Fieldtype Fieldtype
{
get
{
return type;
}
}

public virtual Classdescriptor.Fieldtype[] Fieldtypes
{
get
{
return new Classdescriptor.Fieldtype[]{type};
}
}

public Type Keytype
{
get
{
return mapkeytype(type);
}
}

public object[] this[object from, object till]
{
get
{
return Get(from, till);
}

public object this[object key]
{
get
{
```

```
return Get(key);
}
set
{
Set(key, value);
}
}
protected virtual Key checkkey(Key key)
{
if (key != null)
{
if (key.type != type)
{
                                throw                                new
Storageerror(Storageerror.Errorcode.INCOMPATIBLE_KEY_TYPE);
}
if ((type == Classdescriptor.Fieldtype.tpobject
|| type == Classdescriptor.Fieldtype.tpoid)
&& key.ival == 0 && key.oval != null)
{
object obj = key.oval;
   key    =    new    Key(obj,    Storage.Makepersistent(obj),
key.inclusion != 0);
}
 if (type == Classdescriptor.Fieldtype.tpstring && key.oval is
string)
{
    key     =     new     Key(((string)key.oval).Tochararray(),
key.inclusion != 0);
}
return key;
}

public virtual object Get(Key key)
{
key = checkkey(key);
if (root != 0)
{
Arraylist list = new Arraylist();
 Btreepage.find((Storageimpl) Storage, root, key, key, this,
height, list);
if (list.Count > 1)
{
throw new Storageerror(Storageerror.Errorcode.KEY_NOT_UNIQUE);
}
else if (list.Count == 0)
{
return null;
}
else
{
return list[0];
```

ДОДАТОК Б

Слайди презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Атестаційна робота магістра

Дослідження методів аналізу продуктивності СКБД

Керівник
проф. Лєсна Н.С.

Виконав
ст. гр. ІПЗм-18-2 Чачанідзе С.С.

1

Мета роботи

- Метою роботи є збільшення продуктивності кластерних індексів у СКБД.
- У магістерській роботі досліджується ефективність застосування небінарних дерев ван Едме Боаса для представлення кластерного індексу в порівнянні із традиційними методами з погляду продуктивності.
- Об'єктом дослідження є процес підвищення продуктивності роботи СКБД із використанням кластерних індексів

2

Аналіз методів збільшення продуктивності кластерних індексів

У більшості випадків, правильність індексу визначають дві характеристики: архітектура бази даних і логіка використання бази даних.

Можна скористатися наступними правилами для побудови правильних індексів:

- практично всі таблиці повинні мати первинний ключ, який будується на індексі, особливо якщо цей ключ часто використовується;
- маленькі таблиці (менш ніж 100 рядків) не мають потреби в індексах.
- варто індексувати поля, які часто використовуються в запитах, і не варто індексувати стовпці, які використовуються рідко;
- будь-яке поле, використовуване в функції, що агрегує (сума, агрегат і т.ін.), яка містить пропозиції `GROUP BY` або `ORDER BY` і використовується в JOIN, повинно розглядатися як кандидат на індекс.

3

Існує два типи індексів: кластерні і некластерні.

- При наявності кластерного індексу рядка таблиці впорядковані за значенням ключа цього індексу.
- Якщо в таблиці немає кластерного індексу, таблиця називається купою.
- Некластерний індекс, створений для такої таблиці, містить тільки покажчики на записі таблиці.

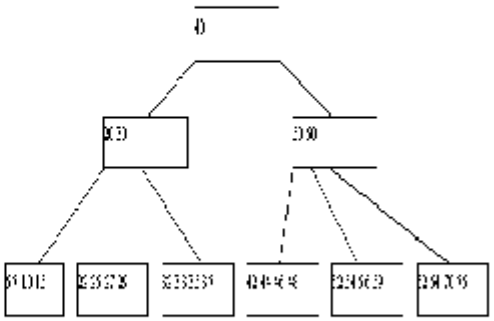
- Кластерний індекс може бути тільки одним для кожної таблиці, але кожна таблиця може мати трохи різних некластерних індексів, кожний з яких визначає свій власний порядок проходження записів.

4

Використання B+ дерев для побудови індексів

- Сімейство B-Tree індексів – це найбільш часто використований тип індексів, що організовані як збалансоване дерево упорядкованих ключів.

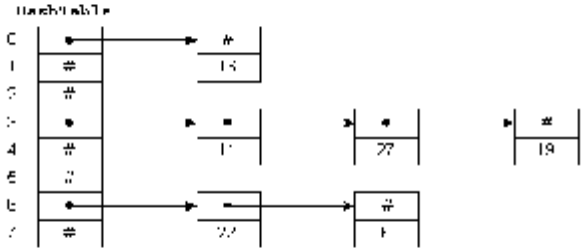
- Вони підтримуються практично всіма СКБД як реляційними, так нереляційними, і практично для всіх типів даних



Приклад будови B-дерева

5

Структура хеш-таблиці



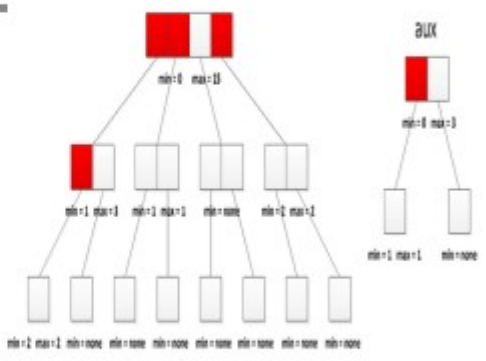
•У багатьох випадках наведені дерева мають незадовільну продуктивність при пошуку, вставці, зміні або видаленні даних.
•Заміна структури індексу може дати підвищення продуктивності роботи СКБД.
•Тому досить обґрунтованим і актуальним бачиться застосування дерева ван Едме Боаса для реалізації кластерних індексів у СКБД.

6

Структура дерева ван Едме Боаса

- Дерево ван Едме Боаса (van Emde Boas tree) – асоціативний масив, який дозволяє зберігати цілі числа в діапазоні
- [0; U), де U = 2k, тобто, числа, що містять не більш ніж *k* бітів.

- Головна особливість цієї структури – виконання всіх операцій за час **O(log(log(U)))** незалежно від кількості елементів, які зберігаються в ній.



7

Постановка завдань дослідження

- проаналізувати структури різних кластерних індексів у СКБД і їх вибір для дослідження й порівняння;
- описати застосування дерев ван Едме Боаса до завдання побудови кластерних індексів;

- обосновати вибір методу проведення експерименту;
- провести алгоритмізацію застосування дерева ван Едме Боаса для побудови кластерних індексів;
- розробити ПЗ, що дозволяє провести порівняльний аналіз продуктивності кластерних індексів при використанні дерева ван Едме Боаса;
- визначити результатів програмного експерименту по аналізу продуктивності БД для збору статистичної інформації;
- оцінити ефективність застосування досліджуваного дерева з погляду продуктивності;

8

Аналіз показників продуктивності СКБД із різними структурами кластерних індексів

- У термінології планування експериментів вхідні змінні й структурні допущення, що представляють модель, називаються факторами, а вихідні показники роботи – відгуками.
- Рішення про те, які параметри й структурні допущення вважати фіксованими показниками моделі, а які експериментальними факторами, залежить від цілей дослідження.

- У багатьох випадках фактори можуть носити не тільки кількісний, але і якісний характер.
- Тому значення факторів зазвичай називають рівнями.

- Якщо при проведенні експерименту можна змінювати рівень фактора, то експеримент називається активним, а якщо ні, то пасивним.

9

Запропонований метод виміру кластерних індексів у СКБД

- Для виконання загального плану роботи знадобиться змінити структуру кластерного індексу в існуючої СКБД на дерево ван Едме Боаса.

- Використання відкритого програмного забезпечення дозволить впровадити у вже існуючу базу даних зміну структури кластерного індексу й провести порівняння з незміненою версією БД.

- Виконання запитів до БД буде проводитися через розроблену утиліту – програмне забезпечення, що дозволяє виконувати запити до бази даних і вести підрахунок даних про час виконання запиту й використання пам'яті.

- Метою математичної обробки результатів експерименту є не знаходження дійсного характеру залежності між змінними або абсолютної величини деякої константи, а репрезентація результатів спостережень у вигляді найбільш простої формули з оцінкою можливої погрішності її використання.
- **Таким чином,** виконані ідентифікація й вибір рівнів факторів, обґрунтована загальна схема організації експериментальних досліджень, доведена необхідність проведення серій послідовних експериментів.

10

Розробка алгоритмів для заміни структури кластерного індексу в БД

Алгоритм виконання операції вставки елемента *x* складається з декількох частин:

Якщо дерево порожнє або в ньому міститься єдиний елемент (*min*=*max*), то полям *min* і *max* присвоєно відповідні значення. Записана інформація в *min* і *max* повністю описує стан поточного дерева й задовольняє структурі нашого дерева.

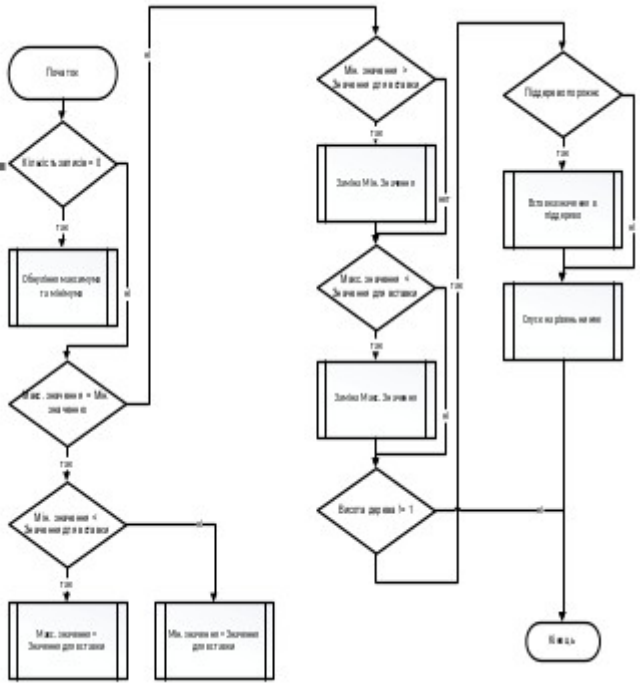
Інакше: Якщо елемент *x* більше *max* або менше *min* поточного дерева, то обновлюється відповідне значення мінімуму або максимуму, а старий мінімум або максимум додається в дерево.

Вставка в допоміжне дерево aux числа high(x), якщо відповідне піддерево children[high(x)] до цього було порожнє.

Вставка числа low(x) у піддерево children[high(x)], за винятком ситуації поточного дерева, – це 1-дерево, і подальша вставка не потрібна (рис.).

11

Схема додавання нового запису



12

Видалення з дерева також поділяється на декілька підзадач

13

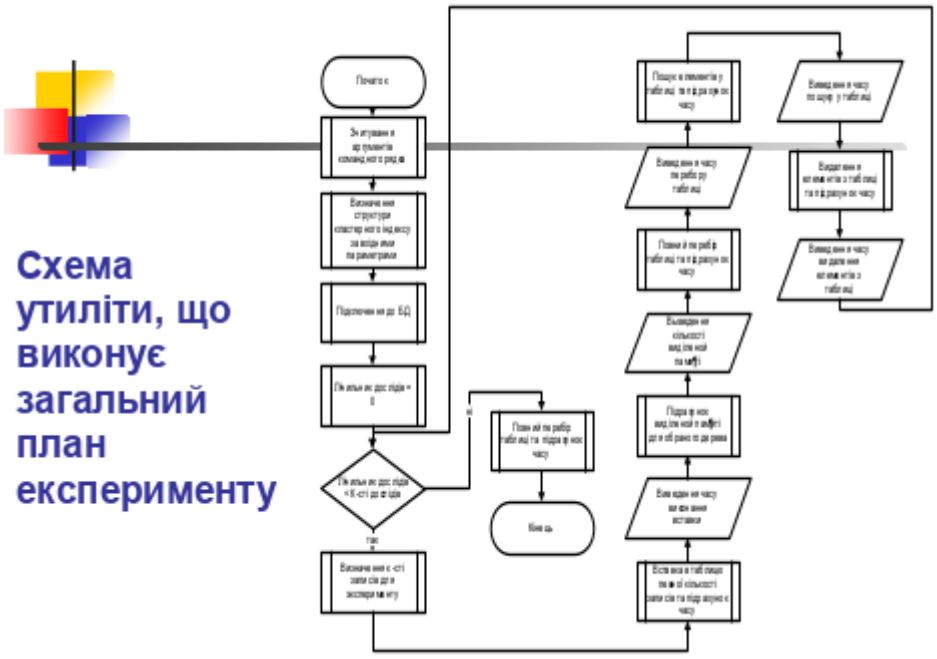


Схема утиліти, що виконує загальний план експерименту

14

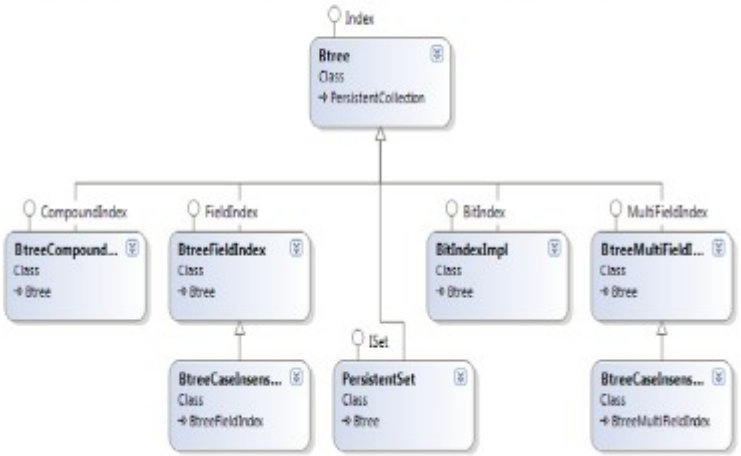
У якості протоколу обміну використовуються виклики функцій (API) бібліотеки Sqlite

- Підхід, що розроблений, зменшує накладні витрати, час відгуку й спрощує програму.
- Sqlite зберігає всю базу даних (включаючи визначення, таблиці, індекси й дані) у єдиному стандартному файлі на тому комп'ютері, на якому виконується програма.

- Простота реалізації досягається за рахунок того, що перед початком виконання транзакції запису, весь файл, що зберігає базу даних, блокується;
- Acid-функції досягаються в тому числі за рахунок створення файлу журналу.

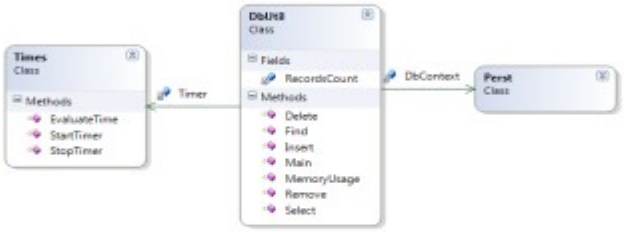
15

Діаграма класів, що реалізують кластерні індекси в БД Perst



16

Діаграма класів додатку, що розроблений



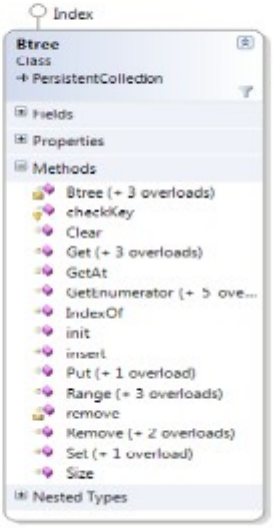
17

Методи в класі `Btree`

Для проведення дослідження не потрібно заміняти всі методи, що описані в класі Btree.

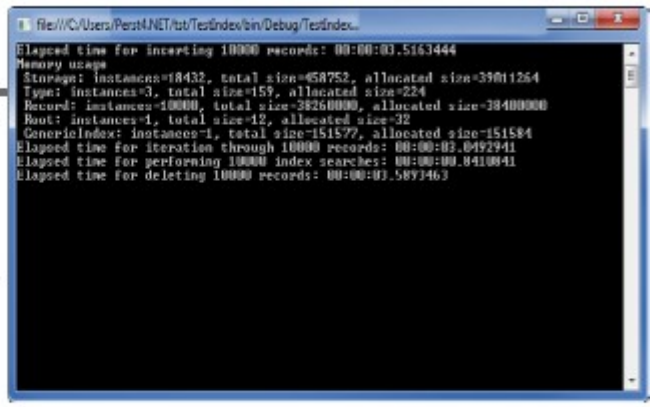
Функціонування індексу можливо при перевизначенні наступних методів:

- `Init()` – ініціалізація дерева;
- `Count()` – повертає кількість елементів у дереві;
- `Getenumerator()` – ітерації по елементах дерева;
- `Get()` – повертає об'єкт по ключу;
- `Find()` – робить пошук по всіму дереву;
- `Insert()` – вставка об'єкта в дерево;
- `Remove()` – видалення об'єкта з дерева.



18

Результати роботи утиліти



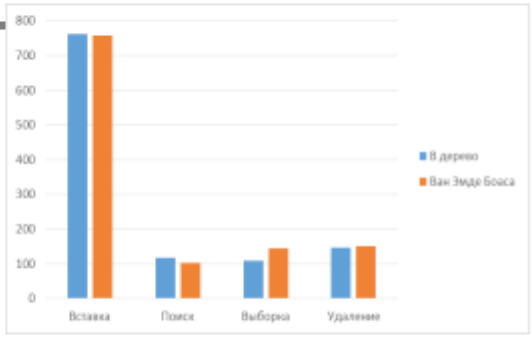
Загальна схема роботи утиліти:
- підключення до БД Perst;
- добуток операції вставки в таблицю з індексом;
- підрахунок пам'яті виділеної для дерева;
- повний перебір записів таблиці;
- пошук випадкових елементів;
- видалення всіх записів таблиці.

При виконанні кожного кроку також буде зроблений вимір часу.

У результаті проведення експериментів по оцінці продуктивності СКБД були отримані відгуки систем для кожного опиту, усього опитів було поставлено 10 для кожного виду СКБД.
- Відгуки представлено в секундах.

19

Час відгуків по кожному дереву для 10000 записів



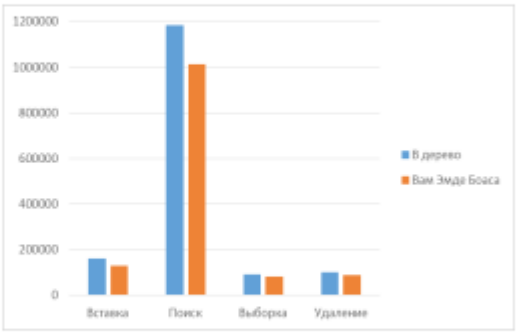
Описано алгоритми для заміни структури кластерного індексу БД.

Представлені наступні алгоритми:
- додавання нового запису в дерево індексу;
- видалення запису з дерева індексу;
- пошук по дереву індексу;
- знаходження наступного елемента;
- одержання максимуму й мінімуму.

- Екранні форми дають підстави зробити висновки, що при використанні таблиць із розміром записів близько 10000 слід використовувати кластерні індекси на основі B-дерев, що дасть можливість і заощадити пам'ять і домогтися високої продуктивності СКБД.

20

Час відгуків по кожному дереву для 1000000 записів



Значення відгуків показали виграш кластерних індексів з використанням дерев ван Едме Боаса в продуктивності при великій кількості даних

21

Висновки

- Розглянуто реалізації структури кластерних індексів та шляхи рішення проблеми збільшення показників продуктивності кластерних індексів у СКБД.

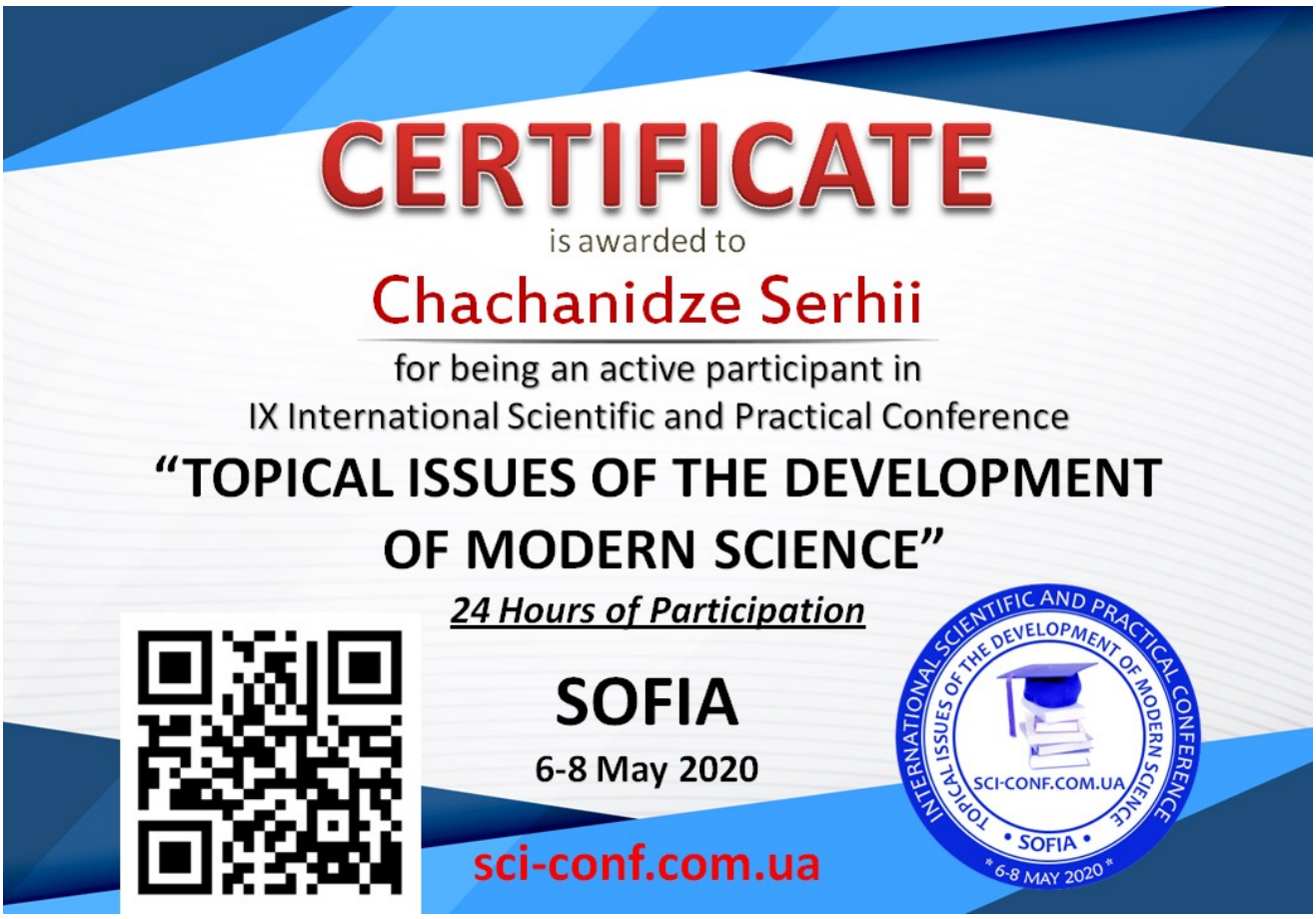
- Найбільш актуальним був обраний спосіб використання іншої структури в основі кластерного індексу – дерева ван Едме Боаса.
- Описані алгоритми для заміни структури кластерного індексу БД.
- Розроблено алгоритми: додавання новому запису в дерево індексу; видалення запису з дерева індексу; пошук по дереву індексу; знаходження наступного елемента; одержання максимуму й мінімуму; службові алгоритми роботи програмного забезпечення для проведення експерименту.
- Спроектований прототип ПЗ, необхідний для проведення експерименту – Treeutil. Обране інструментальне середовище для розробки даного ПЗ.
- Зроблений розрахунок і аналіз математичної моделі та побудовані діаграми, що порівнюють відгуки для різних структур, використаних у кластерному індексі СКБД Perst.

- Значення відгуків показали виграш кластерних індексів з використанням дерев ван Едме Боаса в продуктивності при великій кількості даних.
- По закінченню експерименту й обробки результатів були зроблені висновки по доцільності використання різних структур.
- Розраховано графіки, що порівнюють індекси на підставі різних дерев за експериментальними значенням відгуків, що явно показують виграш дерев ван Едме Боаса в продуктивності при великій кількості записів.

22

ДОДАТОК В

Апробація результатів роботи


CERTIFICATE
is awarded to
Chachanidze Serhii
for being an active participant in
IX International Scientific and Practical Conference
"TOPICAL ISSUES OF THE DEVELOPMENT OF MODERN SCIENCE"
24 Hours of Participation
SOFIA
6-8 May 2020
sci-conf.com.ua
INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE
TOPICAL ISSUES OF THE DEVELOPMENT OF MODERN SCIENCE
SCI-CONF.COM.UA
SOFIA
6-8 MAY 2020