



Projektový seminár 2

FMFI, UK, Pavel Rajčok

Školiteľ

Ing. Alexander Šimko, PhD.

Anotácia diplomovej práce v bodoch (1/2)

- Aby databázový systém fungoval efektívne, je ho potrebné nakonfigurovať.
- Súčasťou tejto konfigurácie je aj vytváranie vhodných indexov, ktoré typicky robí databázový špecialista.
- Vzhľadom aj na finančné náklady je zaujímavé túto činnosť plne automatizovať.
- Problém automatického výberu indexov v databázových systémoch je NP-ťažký a jeho riešeniu bolo venované značne množstvo výskumu.

**Automatický
manažment
indexov
pre PostgreSQL**

Anotácia diplomovej práce v bodoch (2/2)

- Pre komerčné databázové systémy existujú nástroje, ktoré tento problém riešia, pre open-source databázový systém PostgreSQL takéto nástroje chýbajú, s výnimkou veľmi mladého projektu Dexter.
- Cieľom práce je nadviazať na výskum v tejto oblasti, navrhnúť a implementovať nástroj pre automatický výber a manažovanie indexov v open-source databázovom systéme PostgreSQL.
- Súčasťou práce bude aj experimentálne vyhodnotenie relevantných parametrov vytvoreného nástroja.

Automatický
manažment
indexov
pre PostgreSQL



Agenda

01

Problematika

02

Implementácia

Problematika

1




Databázový systém

Databázový systém

- súbor súvislých dát a množiny programov, ktoré nám umožňujú pracovať s nimi
- hlavnou úlohou databázového systému je zapisovať a čítať dáta z databázy
- takéto systémy sú navrhnuté a prispôsobené na správu veľkého množstva informácií
- ide o predefinované štruktúry na ukladanie informácií a mechanizmy pre prístupovanie k nim





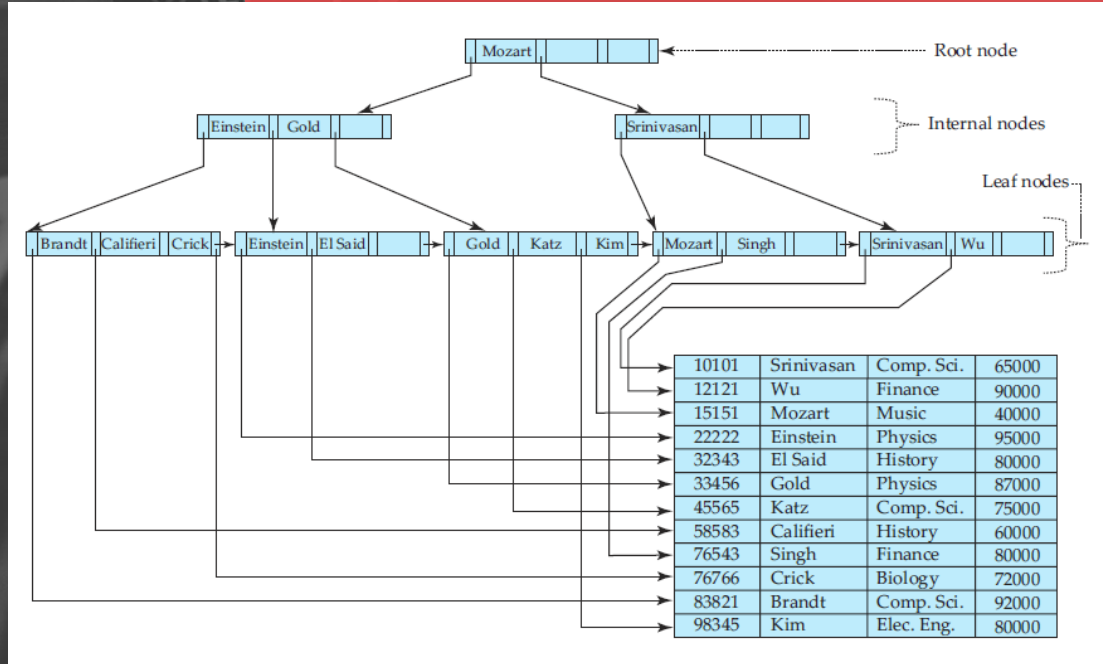
Indexy v databázovom systéme

Index

Indexy zvyšujú výkonnosť databázy a urýchľujú proces hľadania záznamov.

Štruktúra B+ strom

- Dátová štruktúra využívaná v implementácii indexov v databázových systémoch
- Forma vyváženého vyhľadávacieho stromu
- Dĺžka od koreňa ku každému listu je rovnaká
- Každý vrchol obsahuje utriedený zoznam kľúčov a smerníkov, ktoré ukazujú na vrcholy nižšej úrovne



A close-up, shallow depth-of-field photograph of a person's hands typing on a laptop keyboard. The hands have light-colored nail polish. A semi-transparent white circle is centered over the text. To the right of the circle are three red circles of varying sizes, arranged in a cluster. The background is blurred, showing more of the keyboard and the person's hands.

Automatický výber indexov

Automatický výber indexov



1

- Nástroj sa zaoberá záťažou systému
- Analyzuje históriu príkazov typu SELECT a UPDATE.
- História obsahuje množinu záznamov, zaznamenané sú príkazy spomínaného typu počas stanoveného časového úseku

2

- Nástroj vyberie len vhodných adeptov pre ďalší proces – dôjde ku kompresii.
- Množina podobných príkazov bude reprezentovaná jedným všeobecným, ktorý má zhodný charakter a prináleží celkovému počtu volaní tejto množiny.

Automatický výber indexov



3

- Príkazy, ktoré sa vyskytujú v minimálnej miere a nezaťažujú systém, nástroj vynechá.
 - Príkazy, ktoré zaťažujú systém najviac by mali byť uprednostnené na analýzu.
- Kompresia je nevyhnutná hlavne v prípade veľkej zaťaženia systému.

4

- Za pomoci nástroja na optimalizáciu dopytov by mala byť vytvorená množina indexov, ktoré znížia celkové zaťaženie systému.

5

- Riešenie sa následne realizuje za pomoci správneho výberu heuristiky, ktorá vhodným spôsobom znižuje počet možných alternatív vytvorenia indexov.

Problém výberu indexov – ISP problém



- kľúčový problém v návrhu databáz

- problém známy ako NP ťažký

- Pre efektívne fungovanie databázového systému, je potrebné ho nakonfigurovať

- jeho riešeniu bolo venované značne množstvo výskumu

ISP problém

- Minimalizovanie času pri práci s databázou

- pre komerčné databázové systémy existujú nástroje, ktoré tento problém riešia, pre open-source databázový systém PostgreSQL takéto nástroje chýbajú

- Hlavným cieľom je minimalizovať celkový čas vykonávania, definovaný ako súčet časov údržby a odpovedí databázového systému pre všetky dopyty

A close-up, shallow depth-of-field photograph of a person's hands typing on a laptop keyboard. The hands have dark red nail polish. A large, semi-transparent white circle is centered over the keyboard, containing the text 'PostgreSQL'. To the upper right of this circle are three red circles of varying sizes, arranged in a cluster. The background is blurred, showing more of the laptop and the person's arms.

PostgreSQL

PostgreSQL

- voľne šíriteľný objektovo relačný databázový systém
- Vyvinutý na Univerzite Berkeley v Kalifornii
- podporuje množstvo aspektov a vlastností SQL jazyka
- možnosť rozšírenia nových dátových typov, funkcií, metód indexov



PostgreSQL

Implementácia

2

História spustených príkazov



- Rozšírená funkcionálnosť pre PostgreSQL
- Zoskupovanie logov a informácií o nich za určitý časový úsek
- Redukovanie príkazov na základe počtu volaní a časovej náročnosti príkazov

Pg_stat_statements



query text	calls bigint	total_time double precision
select first_name from my_schema.my_table_employees where first_name = ?;	17	0.593983448003356
select e.first_name from my_schema.my_table_jobs j, my_schema.my_table_employees e where e.job_id = j.job_id and e.salary > j.min_salary;	18	3.23376733894988
select * from my_schema.my_table_countries where region_id = ? and country_name = ?;	11	0.23391572790341
select * from my_schema.my_table_countries where region_id = ? and country_id = ? or country_name = ?;	10	0.841583459806051
select * from my_schema.my_table_countries where country_name = ?;	14	0.335264955532492
select * from my_schema.my_table_countries where country_name = ? and region_id = ?;	16	0.416943025056351
select * from my_schema.my_table_countries c where region_id = ?;	14	0.404541642772624

Analyzovanie vnorených príkazov

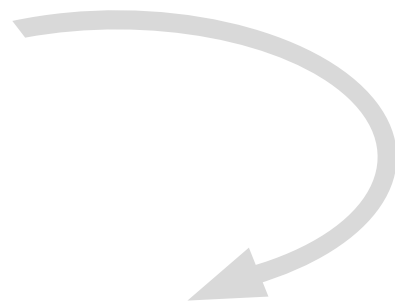


- Potreba rozdelenia príkazu, v prípade, že obsahuje aj vnorené príkazy
- Vytvorenie dátovej štruktúry strom
- Využitie dátovej štruktúry zásobník

```
SELECT *, (SELECT count(*)  
            FROM films AS f2  
            WHERE f2.year < f.year) AS how_many  
FROM films AS f
```



```
while index < len(q):  
    if q[index] == '(':  
        stack.append(index)  
        child = Node()  
        child.setup(node)  
        node.children.add(child)  
        node = child  
    elif q[index] == ')':  
        start = stack.pop()  
        end = index + 1  
        sub_select = q[start+1:end-1]  
        if 'select' in sub_select:  
            index = start  
            q = q[:start] + "?" + q[end:]  
            node.query = sub_select  
        else:  
            node.parent.children.remove(node)  
            node = node.parent  
    index += 1
```



Parsovanie príkazov



- Knižnica pre jazyk Python
- Na základe lexikálnej analýzy vytvorí dátovú štruktúru slovník


MOZ-SQL-PARSER




```
# SELECT * FROM dual WHERE a>b ORDER BY a+b
{
  "select": "*",
  "from": "dual"
  "where": {"gt": ["a", "b"]},
  "orderby": {"add": ["a", "b"]}
}
```

Vytvorenie kombinácií indexov


Na základe rozparsovaného príkazu vytvoríme vhodné kombinácie pre indexy



```
SELECT *  
FROM my_schema.my_table_countries  
WHERE region_id = 2  
      AND country_id = 'AR'  
      AND country_name = 'Argentina'
```



```
('select', '*')  
('from', 'my_schema.my_table_countries')  
('where', {'and': [{'eq': ['region_id', 2]}, {'eq': ['country_id', {'literal': 'AR'}]}, {'eq': ['country_name', {'literal': 'Argentina'}]}]}
```

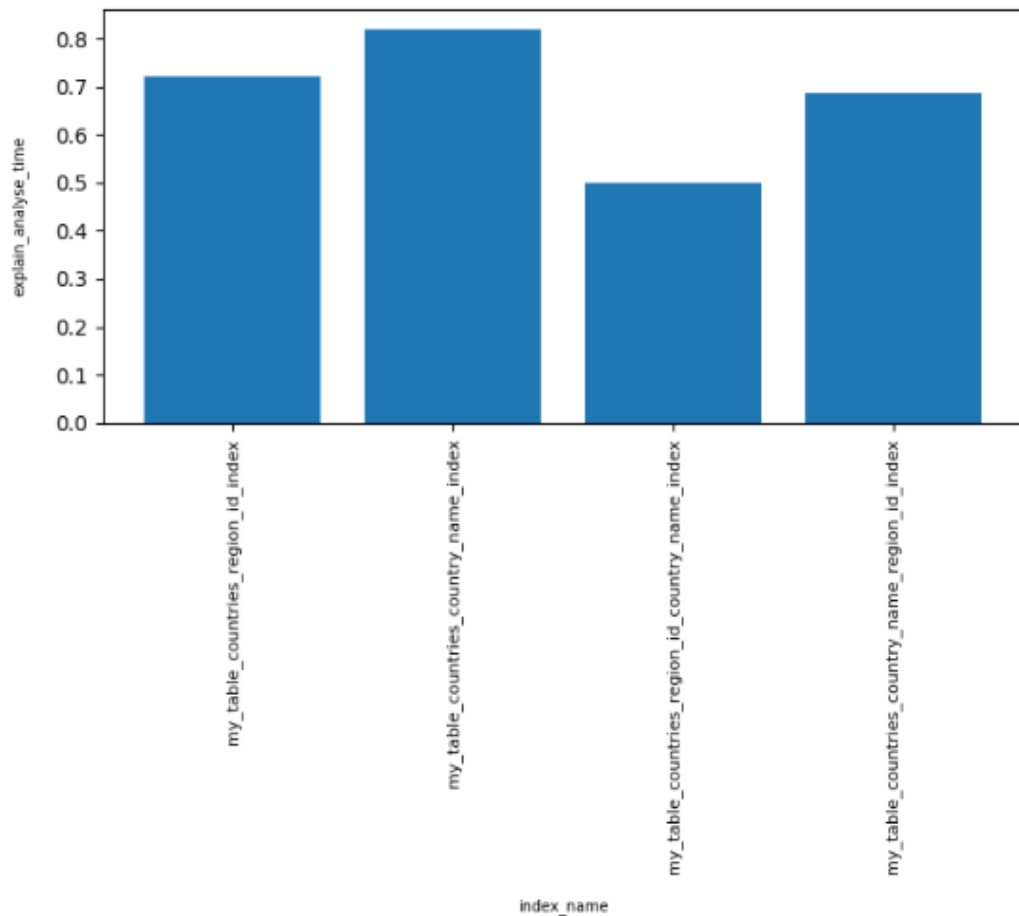


```
*****  
my_table_countries_country_name_index  
*****  
my_table_countries_country_id_index  
*****  
my_table_countries_region_id_index  
*****  
my_table_countries_country_name_country_id_index  
*****  
my_table_countries_country_name_region_id_index  
*****  
my_table_countries_country_id_country_name_index  
*****  
my_table_countries_country_id_region_id_index  
*****  
my_table_countries_region_id_country_name_index  
*****  
my_table_countries_region_id_country_id_index  
*****  
my_table_countries_country_name_country_id_region_id_index  
*****  
my_table_countries_country_name_region_id_country_id_index  
*****  
my_table_countries_country_id_country_name_region_id_index  
*****  
my_table_countries_country_id_region_id_country_name_index  
*****  
my_table_countries_region_id_country_name_country_id_index  
*****  
my_table_countries_region_id_country_id_country_name_index
```


Vytvorenie grafu

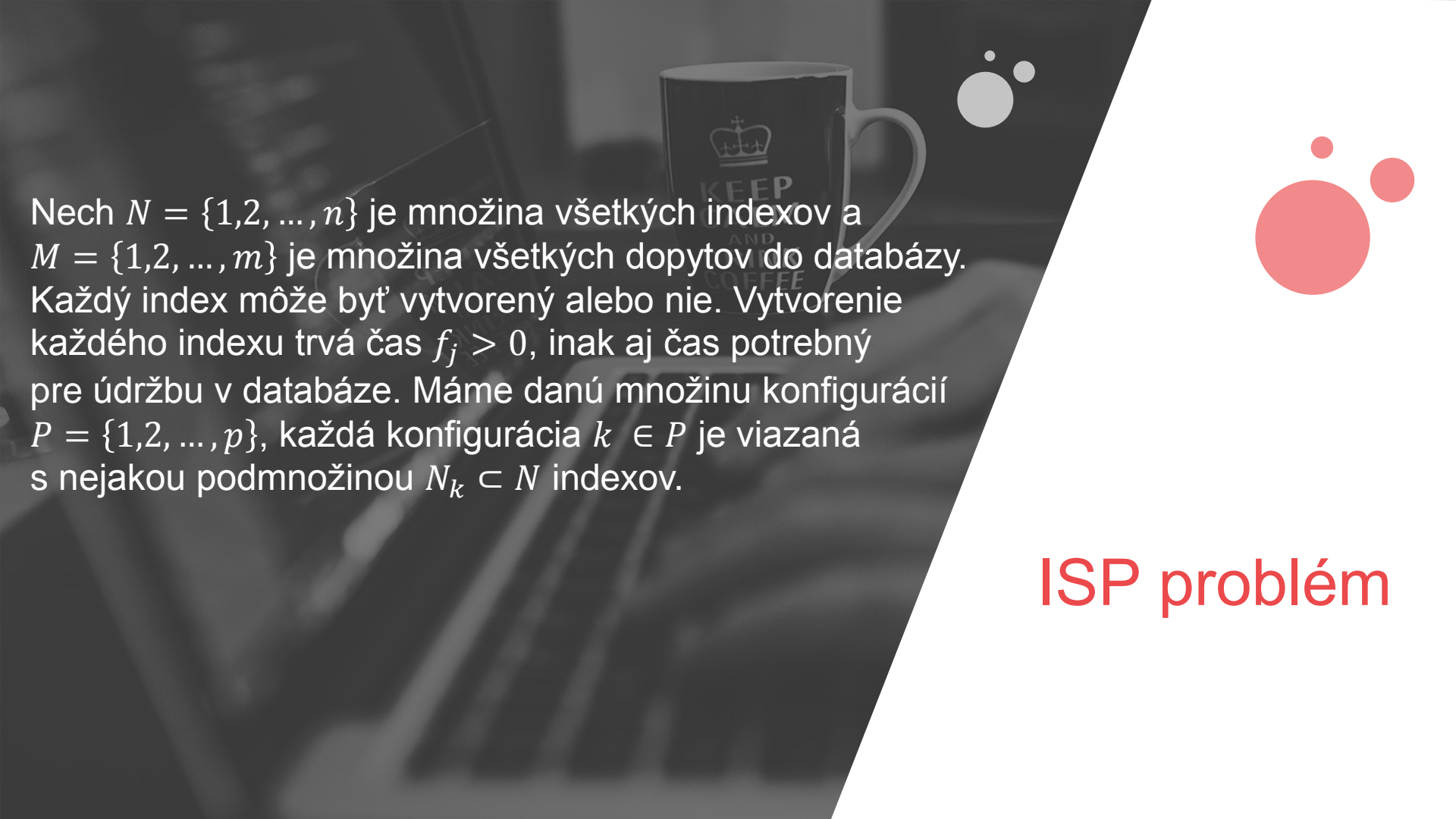
- Použitá knižnica Matplotlib
- Časy vyhodnotené databázovým príkazom EXPLAIN ANALYZE

```
select *  
from my_schema.my_table_countries  
where region_id = 1 and country_name = 'Argentina';
```






Ďalšie kroky
v implementácii



Nech $N = \{1, 2, \dots, n\}$ je množina všetkých indexov a $M = \{1, 2, \dots, m\}$ je množina všetkých dopytov do databázy. Každý index môže byť vytvorený alebo nie. Vytvorenie každého indexu trvá čas $f_j > 0$, inak aj čas potrebný pre údržbu v databáze. Máme danú množinu konfigurácií $P = \{1, 2, \dots, p\}$, každá konfigurácia $k \in P$ je viazaná s nejakou podmnožinou $N_k \subset N$ indexov.


ISP problém



Konfigurácia je aktívna, ak sú všetky jej indexy vytvorené, počas spustenia dopytu $i \in M$, tým získavame $g_{ik} \geq 0$ čas.

V praxi, väčšina párov $(i, k), i \in M, k \in P$ má g_{ik} rovný nule. Toto môže byť jednoducho vysvetlené faktom, že konkrétna konfigurácia má vplyv na obmedzené množstvo dopytov z množiny M . Naším cieľom bude vytvoriť také indexy, ktorých čas potrebný na spustenie všetkých dopytov bude minimalizovaný. T.j. celkový čas g bude maximalizovaný.

ISP problém

A close-up, slightly blurred photograph of a person's hands typing on a laptop keyboard. The person has light-colored skin and is wearing red nail polish. A large, semi-transparent white circle is centered over the text. To the right of the circle, there are three red circles of varying sizes, with the largest one being a solid red circle and the two smaller ones being semi-transparent. The background is a soft, out-of-focus light color.

Ďakujem za
pozornosť