Analytické dotazy a materializované pohledy

Ondřej Benkovský, Tom Bartoň, Andrea Navrátilová a Vojtěch Frnoch

Osnova prezentace

- Popis problematiky
- Cíle projektu
- Představení testovacích dat
- Materializované pohledy v PostgreSQL
- Podpora pro OLAP v PostgreSQL
- Případy užití a výkonnostní testy
- Zhodnocení časového plánu

Popis problematiky

- Analytické dotazy = provádějí komplexní analýzu dat, využití při data miningu
- Využívají často agregační, seskupovací funkce, klouzavé okna, ...
- Provádění takových dotazů může být výpočetně náročné nad neupravenými daty
- OLAP umožňuje uspořádat data v různých dimenzích, tak aby určité analytické dotazy byly efektivnější

Cíle projektu

- Představit podporu materializovaných pohledů v PostgreSQL
- Představit podporu pro OLAP v PostgreSQL
- Implementovat analytické dotazy nad testovacími daty
- Porovnat výkonnost různých implementací analytických dotazů

Představení testovacích dat

- Relace conn_log a service_log
- Relace conn_log, obsahující data o vytvořených spojeních mezi PDA a serverem
- Každé nové spojení vytvoří nový záznam
- Testovací vzorek obsahuje 3710626 záznamů
- Schéma:

```
conn_log(<u>log_key</u>, sim_imsi, time, car_key, pda_imei, gsmnet_id, method, program_ver)
```

Představení testovacích dat

- Relace service_log, obsahující periodická hlášení aplikací
- Aplikace podá hlášení cca každých 10 minut
- Každé nové hlášení vytvoří nový záznam
- Testovací vzorek obsahuje 2323524 záznamů

Schéma:

```
service_log(<u>service_key</u>, car_key, time, app_run_time, pda_run_time, device)
```

Materializované pohledy

- Obdoba SQL view, ale výsledek dotazu je uložen v obdobě tabulky a je uložen i dotaz samotný
- Vytvoření materializovaného pohledu

```
CREATE MATERIALIZED VIEW mymatview AS SELECT * FROM mytab;
```

 Materializovaný pohled není možné přímo upravit (pomocí UPDATE)

Materializované pohledy

Materializovaný pohled lze obnovit pomocí

```
REFRESH MATERIALIZED VIEW mymatview;
```

 Obnovení materializovaného dotazu se provede spuštěním uloženého dotazu

OLAP

- Slouží k rychlé realizaci složitých dotazů
- Použití předpočítaných agregovaných a odvozených hodnot
- Atributy relace rozdělujeme na:
 - Dimenze = udávají "pohled" na agregované atributy (např. časové dimenze den, měsíc a rok)
 - Míry (measure) = atributy, které se agregují (číselné atributy)
- Vytvaření datových kostek uchovávajících předpočítané hodnoty

OLAP příklad

Počet spojení pro verzi v časových dimenzích

dimenze míra

program_ver	year	month	day	conn_count
0.5	2017	1	13	52
0.5	2017	1	14	12
0.5	2017	1		64
0.5	2017	2	1	23
0.5	2017	2		23
0.5	2017			87

 Null hodnota v dimenzi zastupuje agregaci přes danou dimenzi (např. pro atribut day jsou to všechny dny v měsíci)

OLAP v PostgreSQL

- Pro vytváření OLAP kostek jsou potřebné složitější seskupovací akce
- Od PostgreSQL 9.5 přidány operace
 - GROUPING SETS
 - ROLLUP
 - CUBE

GROUPING SETS

SYNTAX

```
SELECT ...

FROM ...

GROUP BY GROUPING SETS( (dimenze), (dimenze), ())
```

- Data jsou seskupena nezávisle dle každé skupiny a následně spojena do jednoho výsledku
- Každá skupina odpovídá jednoduché GROUP BY klauzuli

ROLLUP

SYNTAX

```
SELECT ...
FROM ...
GROUP BY ROLLUP(e1, e2, e3, ...)
```

Je ekvivalentní

```
SELECT ...

FROM ...

GROUP BY GROUPING SETS((e1, e2, e3, ...),

...

(e1, e2), (e1), ())
```

Hierarchický rozpad dimenzí

CUBE

SYNTAX

```
SELECT ...
FROM ...
GROUP BY CUBE (e1, e2, e3, ...)
```

 Ekvivalentní GROUPING SETS nad potenční množinou (e1, e2, e3, ...)

Případy užití

- pro každou verzi programu zjistit počet spojení
- pro každou verzi programu zjistit počty různých zařízení
- pro každé zařízení zjistit počet restartů programu
- pro každé zařízení zjistit kolik hodin bylo používáno
- pro každou verzi programu zjistit počet restartů programu
- který typ zařízení s jakou verzí se nejčastěji restartoval

Pro každou verzi programu zjistit počet spojení

Počet spojení celkem pro každou verzi

```
SELECT program_ver, COUNT(*) as conn_count
FROM conn_log
GROUP BY program_ver;
```

Počet spojení v lednu 2017

```
SELECT program_ver, COUNT(*) as conn_count
FROM conn_log
WHERE time >= '2017-01-01' AND time < '2017-02-01'
GROUP BY program_ver;</pre>
```

Pro každou verzi programu zjistit počet spojení **Výkon**

Počet spojení celkem pro každou verzi

Average planning time: 0.047 ms

Average execution time: 1497.6245 ms

Počet spojení v lednu 2017

Average planning time: 0.07475 ms

Average execution time: 3784.36225 ms

Pro každou verzi programu zjistit počet spojení **Výkon s indexy**

Index nad atributem time v tabulce conn_log

Počet spojení celkem pro každou verzi

Average planning time: 0.047 ms

Average execution time: 1497.6245 ms

Počet spojení v lednu 2017

Average planning time: 0.19475 ms

Average execution time: 1968.90625 ms

Zlepšení o 48%

Pro každou verzi programu zjistit počet spojení OLAP kostka

Dimenze: program_ver, year, month, day

Míry: počet připojení

```
CREATE MATERIALIZED VIEW ver_conns AS

SELECT program_ver,

DATE_PART('YEAR',time) as year,

DATE_PART('MONTH',time) as month,

DATE_PART('DAY',time) as day,

COUNT(*) as conn_count

FROM conn_log

GROUP BY program_ver, rollup(

DATE_PART('YEAR',time),

DATE_PART('MONTH',time),

DATE_PART('DAY',time));
```

Pro každou verzi programu zjistit počet spojení dotazy nad OLAP kostkou

Počet spojení celkem pro každou verzi

```
SELECT program_ver, conn_count
FROM ver_conns
WHERE year is null AND month is null AND day is null;
```

Počet spojení v lednu 2017

```
SELECT program_ver, conn_count
FROM ver_conns
WHERE year=2017 AND month=1 AND day is null;
```

Pro každou verzi programu zjistit počet spojení **Výkon OLAP**

OLAP kostka

Average planning time: 0.08 ms

Average execution time: 21996.8675 ms

Počet spojení celkem pro každou verzi

Average planning time: 0.041 ms

Average execution time: 0.14875 ms

Zlepšení o ~100%

Počet spojení v lednu 2017

Average planning time: 0.08075 ms

Average execution time: 0.143 ms

Zlepšení o ~100%

Pro každou verzi programu zjistit počty různých zařízení

Celkový počet různých zařízení pro každou verzi

```
SELECT program_ver, COUNT(DISTINCT(pda_imei)) as pda_count
FROM conn_log
GROUP BY program_ver;
```

Počet různých zařízení v lednu 2017

```
SELECT program_ver, COUNT(*) as conn_count
FROM conn_log
WHERE time >= '2017-01-01' AND time < '2017-02-01'
GROUP BY program_ver;</pre>
```

Pro každou verzi programu zjistit počty různých zařízení **Výkon**

Celkový počet různých zařízení pro každou verzi

Average planning time: 0.0495 ms Average execution time: 13184.09125 ms

Počet různých zařízení v lednu 2017

Average planning time: 0.0785 ms

Average execution time: 14884.031 ms

Pro každou verzi programu zjistit počty různých zařízení **Výkon s indexy**

Indexy nad atributy time a program_ver v tabulce conn_log

Celkový počet různých zařízení pro každou verzi

Average planning time: 0.1005 ms Average execution time: 7651.78375 ms

Zlepšení o ~41%

Počet různých zařízení v lednu 2017

Average planning time: 0.21 ms

Average execution time: 12070.3565 ms

Zlepšení o ~18%

Pro každou verzi programu zjistit počty různých zařízení OLAP kostka

Dimenze: program_ver, year, month, day

Míry: počet různých zařízení

```
CREATE MATERIALIZED VIEW agg_ver_pda AS

SELECT program_ver,

DATE_PART('YEAR',time) as year,

DATE_PART('MONTH',time) as month,

DATE_PART('DAY',time) as day,

COUNT(DISTINCT(pda_imei)) as pda_count

FROM conn_log

GROUP BY program_ver, rollup(

DATE_PART('YEAR',time),

DATE_PART('MONTH',time),

DATE_PART('DAY',time));
```

Pro každou verzi programu zjistit počty různých zařízení dotazy nad OLAP kostkou

Celkový počet různých zařízení pro každou verzi

```
SELECT program_ver, pda_count

FROM agg_ver_pda

WHERE year is null AND month is null AND day is null;
```

Počet různých zařízení v lednu 2017

```
SELECT program_ver, pda_count
FROM agg_ver_pda
WHERE year=2017 AND month=1 AND day is null;
```

Pro každou verzi programu zjistit počty různých zařízení **Výkon OLAP**

Vytvoření OLAP kostky

Average planning time: 0.20975 ms Average execution time: 46338.722 ms

Celkový počet různých zařízení pro každou verzi

Average planning time: 0.1405 ms Average execution time: 0.10925 ms

Zlepšení o ~100%

Počet různých zařízení v lednu 2017

Average planning time: 0.066 ms Average execution time: 0.1225 ms

Zlepšení o ~100%

Pro každé zařízení zjistit počet restartů programu

• Pomocný view (vybere všechny restarty pro všechny zařízení)

Pro každé zařízení zjistit počet restartů programu

• TOP 10 zařízení, které se restartovaly v lednu 2017

Kolikrát se celkem každé zařízení restartovalo

```
SELECT pda_imei, count(*) as restart_count
FROM view_pda_app_restarts
GROUP BY pda_imei;
```

Pro každé zařízení zjistit počet restartů programu **Výkon**

TOP 10 zařízení, které se restartovaly v lednu 2017

Average planning time: 0.2816 ms

Average execution time: 25440.2064 ms

Kolikrát se celkem každé zařízení restartovalo

Average planning time: 0.2854 ms

Average execution time: 25852.3322 ms

Pro každé zařízení zjistit počet restartů programu **Výkon s indexy**

- Index nad atributem app_run_time v tabulce service_log
- TOP 10 zařízení, které se restartovaly v lednu 2017

Average planning time: 0.513 ms Average execution time: 18968.7646 ms

Zlepšení o ~25%

Kolikrát se celkem každé zařízení restartovalo

Average planning time: 0.5034 ms

Average execution time: 19629.6226 ms

Zlepšení o ~24%

Pro každé zařízení zjistit počet restartů programu **OLAP kostka**

Dimenze: pda imei, year, month, day

Míry: počet restartů

```
CREATE MATERIALIZED VIEW agg pda app restarts AS
SELECT pda imei,
       DATE PART ('YEAR', service log.time) as year,
       DATE PART ('MONTH', service log.time) as month,
       DATE PART ('DAY', service log.time) as day,
       count(*) as restart count
FROM (
      SELECT pda imei,
             car key,
             time as session begin,
             lead(time,1,now()) OVER (Partition by car key ORDER BY time) AS
                                                                           session end
      FROM conn log
      ) t1
      INNER JOIN
      service log
      ON t1.car key = service log.car key
         AND service log.time >= session begin
         AND service log.time <= session end
WHERE app run time <= 0.17
GROUP BY rollup(
                DATE PART ('YEAR', service log.time),
                DATE PART ('MONTH', service log.time),
                DATE PART('DAY', service log.time)
                ), pda imei;
```

Pro každé zařízení zjistit počet restartů programu dotazy nad OLAP kostkou

TOP 10 zařízení, které se restartovaly v lednu 2017

```
SELECT pda_imei, restart_count

FROM agg_pda_app_restarts

WHERE year='2017' AND month='1' AND day is null

ORDER BY restart_count DESC

LIMIT 10;
```

Kolikrát se celkem každé zařízení restartovalo

```
SELECT pda_imei, restart_count
FROM agg_pda_app_restarts
WHERE year is null AND month is null AND day is null;
```

Pro každé zařízení zjistit počet restartů programu **Výkon OLAP**

OLAP kostka

Average planning time: 0.3842 ms

Average execution time: 28380.1018 ms

TOP 10 zařízení, které se restartovaly v lednu 2017

Average planning time: 0.4856 ms

Average execution time: 2.0048 ms

Zlepšení o ~100%

Kolikrát se celkem každé zařízení restartovalo

Average planning time: 0.042 ms

Average execution time: 2.4092 ms

Zlepšení o ~100%

Zhodnocení časového plánu

Task Name	Start Date	End Date	Duration	Assigned To
Nastudování problematiky	03/22/17	03/29/17	20h	Všichni
Analýza testovacích dat	03/27/17	03/29/17	10h	Ondřej
Návrh analytických dotazů	03/30/17	04/05/17	10h	Andrea
Implementace dotazů	04/06/17	04/14/17	15h	Andrea
Implementace dotazů	04/06/17	04/14/17	10h	Tom
Implementace dotazů	04/06/17	04/14/17	6h	Ondřej
Validace dotazů	04/15/17	04/19/17	10h	Tom
Testování výkonnosti dotazů	04/20/17	04/24/17	8h	Vojtěch
Vyhodnocení výkonnosti	04/25/17	04/28/17	10h	Vojtěch
Příprava finálního výstupu	04/29/17	05/04/17	10h	Ondřej

Mar			Apr			May					
Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	We
		Nas	tudování problematiky								
		Ana	lýza testovacích dat								
			Návrh ar	alytických dotazů							
				Imp	lementace dotazů						
				Imp	lementace dotazů						
				Imp	lementace dotazů						
					Validace	dotazů					
						Testování výl	konnosti dotazů				
						Vył	nodnocení výkonn	osti			
							Přípra	va finálního výstu	pu		