

Záludnosti PostgreSQL

Jakub Ouhrabka

Setkání CSPUG

19.4.2011

Něco o mně

- Jakub Ouhrabka
- ComGate Interactive, s.r.o.
- PostgreSQL od verze 7.0 v roce 2000

Nepoužívat SELECT FOR UPDATE

- SELECT ... FOR UPDATE LIMIT 10;
- Standardně READ COMMITTED ISOLATION LEVEL
- FOR UPDATE se provede až nakonec, tzn. SELECT může vrátit záznam, který neodpovídá WHERE podmínce (kdyby ho jiná transakce mezitím změnila)
- Řešení
 - Změnit ISOLATION LEVEL na SERIALIZABLE (9.1)
 - Advisory lock (9.1)
 - UPDATE zamky SET a = 1 WHERE zamek = 'uzivatel-zmena';
 - LOCK TABLE zamykaci_tabulka;

Nepoužívat SERIALIZABLE

- Nepoužívat
- Nefunguje dle očekávání (neplatí pro 9.1)
- Pozor na ROLLBACKy

Nepoužívat hash indexy

- Nejsou logované do WAL
- Výkon není o moc vyšší
- Nепrobíhá na nich vývoj (pár výjimek)
- Nejsou tolik otestované

Nepoužívat RULE

- V některých případech nepředvídatelné chování
- Vývojáři PostgreSQL uvažují o jejich zrušení
- Pokud už je to nutné, tak TRIGGER, ale i s tím šetřit

Nepoužívat/omezit použití TRIGGERŮ

- Zpracování fronty TRIGGERŮ
 - dříve asi(?) bylo $O(n^2)$
 - problém i teď, pokud se fronta nevejde do paměti
- Nečitelný kód, „vedlejší efekt“
- Preferovat pl/pgSQL

Nepoužívat dědičnost

- Vhodná pouze na partitioning, ale jen když doopravdy musím
- Nejde udělat index
- Nefungují foreign keys
- Ne úplně efektivní dotazy
- Vylepšení v 9.1: merge append

- `SELECT * FROM transakce ORDER BY id DESC LIMIT 100;`

Nepoužívat BYTEA, ARRAY v PHP

- Nepodporují ovladače
- Nefunguje 100% správně (znaky s kódem 0, ...)

Deadlocky FOREIGN KEYS

- Použití příliš silných zámků

```
CREATE TABLE foo (a int PRIMARY KEY, b text);  
CREATE TABLE bar (a int NOT NULL REFERENCES foo);  
INSERT INTO foo VALUES (42);
```

```
SESSION 1  
BEGIN;
```

```
INSERT INTO bar VALUES (42);
```

```
SESSION 2  
BEGIN;
```

```
UPDATE foo SET b = 'Hello World' ;
```

http://www.commandprompt.com/blogs/alvaro_herrera/2010/11/fixing_foreign_key_deadlocks/

- Řešení
 - Zamykat konkurentní přístup
 - Připravit aplikaci na deadlocky
 - Nepoužívat cizí klíče
 - 9.2 – možná, částečně...

psql, ruční dotazy

- Nenechávat dlouho trvající transakce
 - IDLE in transaction
 - Vyhledávat!
 - `statement_timeout`
 - Ovlivňuje všechny databáze v clusteru (vacuum apod)
- psql
 - Nedělat:
 - `BEGIN; UPDATE xxx ...;`
 - Použít:
 - `BEGIN;`
 - `UPDATE xxx ...;`

Nastavení prostředí

- Pozor na locale (master, slave cluster)
- Pokud jiná definice locale, dotazy používající indexy mohou vracet špatné výsledky
- Pozor na změnu času letní/zimní

Čeština

- Fulltext – dlouhá inicializace velkých ispell slovníků
 - vyřešeno 9.1
 - jinak pgbouncer
- to_ascii
 - `select to_ascii(convert('Příliš žlutý kůň' using utf8_to_iso_8859_2),'latin2');`
 - nefunguje úplně dobře, nebere v potaz znaky mimo ISO 8859-2
 - lépe použít `regexp_replace(translate(...))`
- Regulární výrazy `[:alpha:]` „nepozná“ znaky s háčky, čárky
 - teoreticky by mělo fungovat
 - přetížení operátoru: ... `replace(op_left, '[:alpha:]', 'aábcčdd'...)` „Vrať 10 nejbližších ...
- Neměnit katalogy – `pg_upgrade`

PL/pgSQL

- Jiný plán než v psql
- Zjistit plán přes
 - psql PREPARE, EXPLAIN EXECUTE
 - plpgsql FOR x IN EXPLAIN SELECT ... LOOP RAISE NOTICE
- Při změně statistik nedochází k přeplánování
 - lze vynutit pomocí `recompile_all_functions` (`ALTER FUNCTION RENAME`)
- Nepoužívat procedury stejných názvů s různými parametry

SQL doporučení

- Kde to jde, tak NOT NULL – „divné“ chování NULL
- Nepoužívat CHAR – chová se „divně“, typicky chci TEXT
- Nepoužívat JOIN USING, NATURAL JOIN
- Nepoužívat UNION, typicky chci UNION ALL
- Nepoužívat float/double, typicky chci NUMERIC
- Nepoužívat TIMESTAMP, typicky chci TIMESTAMPTZ
- Pozor na operace s intervalem, přechod měsíce, změna zimní/letní čas

Záludnosti PostgreSQL

Dotazy?

Děkuji za pozornost!

jakub.ouhrabka@comgate.cz